sensing,
interaction &
perception lab

Department of Computer Science, ETH Zürich

Master's Thesis

Heart Rate Estimation from Wrist-Worn Accelerometers While Asleep

Lars Hauptmann

1st supervisor

Prof. Dr. Christian Holz
Sensing, Interaction & Perception Lab
ETH Zürich

2nd supervisor

Max Möbus
Sensing, Interaction & Perception Lab
ETH Zürich

June 3th, 2024

Lars Hauptmann
Heart Rate Estimation from Wrist-Worn Accelerometers While Asleep
Master's Thesis, June 3th, 2024
Supervisors: Prof. Dr. Christian Holz and Max Möbus

ETH Zürich
Department of Computer Science, ETH Zürich
Universitätstrasse 6
8092 Zürich

Heart Rate Estimation from Wrist-Worn Accelerometers While Asleep

Lars Hauptmann
Sensing, Interaction & Perception Lab
Department of Computer Science, ETH Zürich
lhauptmann@ethz.ch

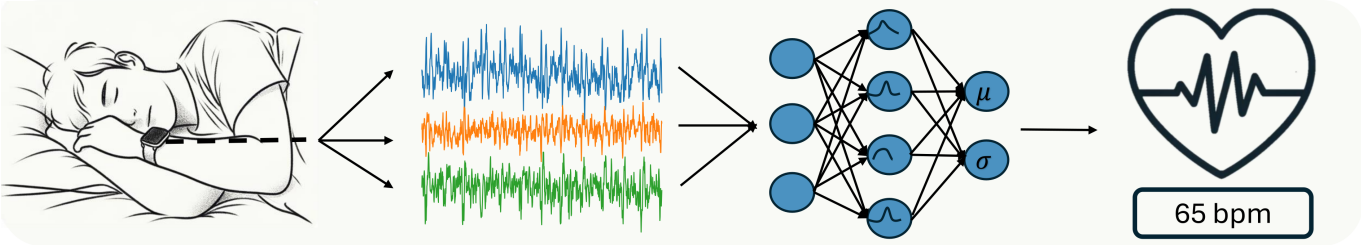

Figure 1: A wristband sensor records accelerometer signals from participants while asleep. This signal is processed and used to train a deep-learning model. The deep learning model predicts the instantaneous heart rate from the accelerometer signal.

ABSTRACT

The development of large biomedical datasets with integrated activity-tracking data has facilitated novel methodologies for heart rate estimation using accelerometry. In this study we leverage accelerometer signals, captured from wrist-worn devices during sleep, which are indicative of ballistocardiographic (BCG) movements. We systematically analyze these signals through signal processing and deep learning techniques to estimate heart rates accurately. We establish a deep learning baseline model and enhance it by integrating self-supervised learning (SSL), uncertainty estimation, and post-processing methods. The evaluation involves training and testing the model on two real-world datasets, comprising 62 participants in total. The refined model demonstrates an improvement over both the baseline and traditional signal processing approaches, accurately predicting heart rates within 5 bpm of the ECG and PPG-derived ground truth for 77 % of the estimates. It achieves an average mean absolute error (MAE) of 2.96 bpm and an average Pearson's correlation coefficient of 0.76 on both datasets. Detailed analyses reveal that the model is sensitive to the frequencies associated with the BCG and the respiratory signal for its prediction. Although these results are promising, further validation on larger and more demographically diverse datasets is required to generalize these findings. This research highlights the potential for extracting heart rates in large-scale biomedical datasets for further analysis of health-related insights.

Keywords

Ballistocardiography; Heart Rate Estimation; Neural Networks; Self-Supervised Learning; Uncertainty Estimation

INTRODUCTION

Wearable technology is becoming a part of daily life for an increasing number of people. Modern-day wearable technology can track health-related signals such as activity, heart rate, blood pressure, and blood oxygenation. This capability is one reason why large biomedical databases are adopting wearable technology to enhance research quality and scope. A notable example is the UK Biobank [18], which equipped 100,000 participants with wrist-worn accelerometers for a week to track their physical activity continuously. This large-scale collection of real-time physiological data allows researchers to analyze daily activity patterns in relation to various health outcomes, such as type 2 Diabetes [58], Parkinson's disease [104], and Depression [15]. Moreover, it has been used to estimate sleep parameters such as the sleep period time [101].

These applications motivate further investigations into which health-related parameters can be estimated solely from wrist-worn accelerometers. Heart rate estimation, for instance, utilizes the Ballistocardiography (BCG) effect, where the mechanical impact of the heart's activity is detected through bodily reactions to the blood surge with each heartbeat. Although this effect is predominantly recorded in stationary settings, it can also be captured by sensitive accelerometers in wearable devices. This capability allows for continuous monitoring of heart rate [46, 94, 116], which can also aid in sleep stage estimation [29].

Although less accurate than state-of-the-art techniques like Electrocardiography (ECG) and Photoplethysmogram (PPG), a BCG-based heart rate estimation would provide considerable statistical power when only accelerometer traces are available, such as in the UK Biobank dataset. By focusing primarily on

sleep periods, where user movement is minimized, the potential for successful heart rate estimation increases significantly, enhancing the potential of health data derived from wearable devices.

Challenges such as motion artifacts (MA) and variations in real-world conditions still pose significant hurdles to obtaining precise heart rate estimates from accelerometer data. Additionally, the scarcity of large-scale datasets impedes the training of robust deep-learning models. A potential solution could involve using self-supervised learning (SSL) with large datasets from related domains that lack heart rate information. Another approach could be to develop methods for estimating the uncertainty of heart rate estimates to identify and correct inaccuracies, leveraging the temporal continuity of heart rate data.

In this thesis, we present an extensive evaluation of techniques for wrist-worn BCG-based heart rate estimation while asleep. After extensive literature research, we implement and compare multiple signal processing-based and deep learning-based approaches on two datasets. We improve the initial models further by using self-supervised learning (SSL) and uncertainty estimation, together with probabilistic postprocessing. The resulting model reliably predicts the heart rate on the given dataset for sleeping subjects, achieving an average MAE of 2.96, and an average Pearson's correlation coefficient of 0.76 on both datasets. Further analysis is conducted, analyzing the sensitivity and qualitative performance of the model and showing that the model acts similarly to frequency-based approaches when estimating the heart rate.

The key contributions of this work are:

- A thorough review of different techniques for wrist-worn BCG heart rate estimation, including techniques from similar domains such as PPG and human activity recognition.
- The development and validation of the first deep-learning model for wrist-worn BCG heart rate estimation employing self-supervised learning (SSL) and uncertainty-based postprocessing, outperforming the signal-processing by 40 % in terms of MAE.
- Interpretation of the implemented deep-learning model and testing against different signal-processing-based models.

RELATED WORK

Ballistocardiography

Ballistocardiography (BCG) is a measurement of the recoil forces of the body as a reaction to the cardiac ejection of blood [91]. The BCG effect was first discovered in 1877 by Gordon [35]. It was first measured in 1905 by Yandell Henderson using a "swinging table" and a set of levers [50]. Subsequently, different techniques were developed to measure the BCG signal. In 1939, Starr et al. used a complex apparatus consisting of a mobile top table, a camera, and multiple mirrors to create repeatable measurements of the BCG signal [91].

The BCG signal is caused by blood traveling along the vascular tree and, therefore, changing the body's center of mass, following Newton's third law. It can be measured as a displacement, velocity, or acceleration and has components on all three axes. These axes are most commonly referred to as longitudianal (head-to-foot), transvere (side-to-side), and dorsoventral (anterior-to-posterior) [85, 50].

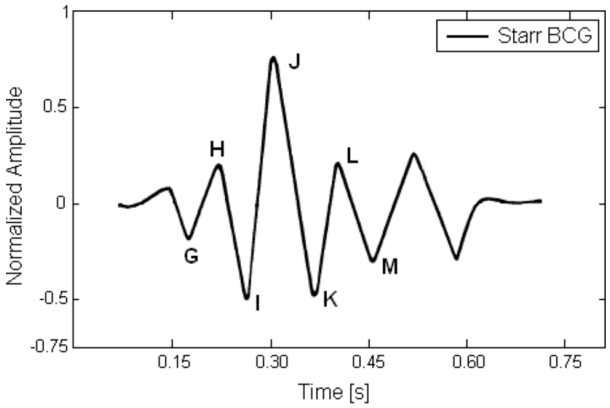

Figure 2: Depiction of a typical BCG waveform, adopted from [75, 92]. The different components are shown with their corresponding letters G,H,I,J,K,L,M.

The BCG waveform, generated with each heartbeat, exhibits distinctive components reflecting specific events in the cardiac cycle. As defined by Starr and Schroeder [92], the components are G, H, I, J, L, M, see Figure 2. These components are grouped into a pre-systolic group, a systolic group, and a diastolic group [75]. Each component is associated with mechanical events during a period of a heartbeat, described in detail by Pinheiro et al. [75].

The G wave is associated with atrial contractions and systole, respectively, visible during conditions like slow heart rates and bradycardia. The H Wave signifies the beginning of ventricular contraction, reflecting a rise in intraventricular pressure. The I Wave indicates the start of the systolic ejection, aligning with key heart sounds. J and K Waves depict the dynamics of ventricular ejection and the end of systole, linked to blood acceleration and deceleration influenced by arterial resistance. L and M Waves occur during diastole, relating to the opening of the atrioventricular valves and aortic dynamics, representing intricate forces within the circulatory system during the heart's relaxation phases. [75]

Since the BCG amplitude is mainly determined by cardiac output, cardiac force, and the velocity of the ejection [75], a potential application of BCG measurement is to estimate the cardiac output [91]. The waveform is also affected by a patient's age, being a good estimator of the heart's age [93]. Other conditions, such as acute myocardial infarction, angina pectoris, and asymptomatic coronary artery disease, alter the BCG waveform and are an early indicator of future complications [75]. BCG technology could, therefore, help to recognize these earlier. Steffensen et al. [94] showed that both heart rate and BCG amplitude increase and pulse transit time decreases with exercise load [75].

Modern BCG measurement systems employ different approaches, primarily focusing on longitudinal (head-to-foot) or dorsoventral (anterior-to-posterior) axes. Since gravity and contact with other objects, such as the floor, interfere with the measurements, they are primarily measured along the gravitational axis. Hence, longitudinal measurements are commonly obtained using force plates or force sensors placed on weighing scales or under the seat of a chair, with the subject in a vertical position. Dorsoventral measurements typically involve the subject in a horizontal position, for example, with pressure sensors under the bed mattress [50]. Bed BCG has been successfully used to estimate sleep stages, utilizing that the momentary heart rate is associated with the sleep stage [55]. Another common measurement of mechanical forces, resulting from the vascular system's recoiling forces, is called Seismocardiography (SCG), where an acceleration sensor is placed on the chest, measuring the local vibrations evoked by the heartbeat [83].

According to Yao et al. [107], who asked 22 subjects to stand still on a force plate and recorded their BCG-signal, the frequency of the fundamental harmonics range from 2.9 Hz to 7.8 Hz, which includes the 2nd to the 7th harmonic. They recommended a pass-band filter for heart rate estimation, and the cut-off frequency should be outside of 3 Hz to 10 Hz.

While BCG-based systems mainly operate in a stationary setting, devices have been developed to measure the BCG-signal in mobile settings as well [44, 46, 45, 94, 68, 68, 116]. These studies mainly operate on wrist- or head-worn devices. Hence, the sensed signal might no longer be aligned with the gravitational axis. Moreover, these approaches operate in a mobile setting and rely on the subject to stay still during the measurement. Since motion happens in the same physical domain as BCG-measurement, motion artifacts overlay the measured signal directly and make detecting a BCG-signal very difficult.

Photoplethysmography

Photoplethysmogram (PPG) is a non-invasive optical technique to detect blood volume changes in tissue microvasculature. The PPG waveform comprises a pulsatile component synchronized with each heartbeat, known as the 'AC' component, and a slower varying 'DC' baseline influenced by factors such as respiration, sympathetic nervous system activity, and thermoregulation. Despite not fully understanding the origins of the components, it is accepted that PPG can provide valuable cardiovascular information. The technique's simplicity, low cost, and portability have led to a resurgence of interest, with applications spanning from oxygen saturation measurement to assessing autonomic function and peripheral vascular disease. [1]

Researchers have explored the complex interaction of light with biological tissue, highlighting factors such as scattering, absorption, and reflection. Optimal wavelength selection, typically in the red or near-infrared range, allows for improved tissue penetration and reduced interference from water and melanin absorption. Pulse wave analysis techniques extract both AC and DC components for further analysis, providing insights into the capillary nutritional blood flow and thermoregulatory blood flow through arterio-venous anastomosis shunt vessels. [1]

A modern-day PPG device comprises an LED and a matched photodetector device. A high-pass filter reduces the DC component of the received signal and amplifies the AC component. There are PPG devices based on transmission and reflection. Clinically, PPG finds wide application in physiological monitoring, vascular assessment, and autonomic function analysis. Commercially available medical devices leverage PPG technology for measuring parameters like oxygen saturation, blood pressure, and cardiac output. Additionally, PPG enables assessment of autonomic function and detection of peripheral vascular disease. The technique's simplicity and compatibility with low-cost semiconductor components make it suitable for primary care and community-based clinical settings, driving its adoption in diverse medical applications.

However, also the PPG-signal is prone to noise and motion artifacts, which makes it difficult to derive clinical diagnosis. Much work has been dedicated to assessing the quality of the PPG-signal [73]. Generally, a low-noise and low-motion signal can be used for clinical diagnosis, while with the onset of noise, only heart rate estimation remains possible, and with the onset of motion artifacts, no analysis is possible [22]. A common way to determine the presence of motion artifacts is to analyze the simultaneously recorded acceleration signal. [1]

Heart Rate Estimation from Ballistocardiography and Photoplethysmography

Heart rate (HR) describes the frequency of the cardiac cycle, consisting of myocardial contraction and relaxation. It not only reflects the cardiovascular system's status but also indicates the state of autonomous nervous system activity and metabolic rate [110]. The heart rate is measured in beats-per-minute [bpm] and equals the average number of cardiac cycles in one minute. A human's average heart rate is 79.1 with differences in age, gender, ethnic group, lifestyle, and weight [3]. Resting heart rate is the heart rate while sitting, sleeping, or in a lying position. It is associated with cardio-respiratory fitness [33] and cardiovascular mortality [26].

The HR can be measured by different means, for example by ECG, PPG, and BCG. Realizations of these approaches also exist contactless [105]. The American National Standards Institute's standard for cardiac monitors, heart rate meters, and alarms defines an accurate HR-measurement as a "readout error of no greater than ±10 % of the input rate or ±5 bpm, whichever is greater." [24]. Generally, ECG is accepted as a gold standard for heart rate measurement. However, extracting the heart rate from ECG signals requires an algorithm to detect the R peaks in the signal. Moreover, it can be affected by motion artifacts. Therefore, some heart rate estimation algorithms additionally estimate the HR estimation quality based on the success of the R peak detection [66].

Signal Processing Approaches for Heart Rate Estimation

Estimating the heart rate from either PPG or BCG has mostly been done with classical signal processing algorithms. Although both signal types are fundamentally different, similar approaches have been used. Both signals are represented by an

input signal with similar sampling frequencies, and both signals can be processed in either the frequency- or time-domain. Both signals have an underlying heart-rate-dependent component and are disturbed by motion artifacts. However, while the PPG signal is only resolved in time-dimension, the BCG signal is generally resolved in three additional dimensions representing orthogonal axes. Also, the PPG is sparse in the frequency domain, whereas the BCG signal has many more different components in the frequency domain. Further details are presented in Table 1.

Photoplethysmogram-based Heart Rate Estimation

With broad applications and an increasing presence of wrist-worn sensing modalities, great interest has been taken in developing methods for PPG heart rate estimation. In particular, the IEEE Signal Processing Cup 2015 [113] has brought popularity to the task. It consists of a dataset with two-channel PPG signals, three-axis acceleration signals, and one-channel ECG signals simultaneously recorded from subjects performing different activities like running and walking. Most methods are evaluated on the training set, where 12 subjects are recorded while running on a treadmill at various speeds. Generally, most methods can be split into signal filtering, heart rate estimation, and postprocessing.

The filtering steps mainly aim to remove motion artifacts and extract a clean signal with dominant peaks for each heartbeat. It usually starts with a bandpass filter, downsampling, and z-normalization as a preprocessing step. Following that, the approaches can be separated into adaptive filtering [108, 67, 74], independent component analysis (ICA) [54], frequency domain ICA [56], wavelet-based denoising [53, 97], singular spectrum analysis (SSA) [113, 112], and other decomposition methods [106], spectral subtraction [82, 28], and Kalman filtering [59, 5]. TROIKA [113] decomposes the signal using SSA and then selects the components that do not share frequency peaks with the acceleration signal. Lakshminarasimha et al. [69] remove motion artifacts using adaptive noise cancellation, whereas Temko et al. [100] apply a Wiener filter to remove the motion noise from the signal. Boloursaz et al. [67] remove motion artifacts iteratively by eliminating components from the PPG signal that correspond to accelerometer components with frequencies between 0.4 Hz and 5 Hz.

The heart rate extraction step finds an estimate of the instantaneous heart rate for each window using the filtered signal. Since for the IEEE Signal Processing Cup 2015 the window size is 8 seconds, and the sampling rate is 125 Hz, 1000 samples can be used for the spectrum estimation. Using a fast fourier transform (FFT), a frequency resolution of 0.1 Hz can be achieved, corresponding to a heart rate resolution of 6 bpm. To overcome this limitation, different approaches either use a high-resolution spectral estimation algorithm [113, 112, 67] or methods like phase vocoders [100] and averaging techniques [115]. TROIKA [113] reconstructs the spectrum of the PPG signal with a basis of harmonics, assuming that the solution is sparse and using the FOCUSS [36] algorithm. The JOSS [112] framework extends this idea by reconstructing the PPG and accelerometer signal together, assuming that the signals share specific frequency components. Zhao et al. [115] uses a moving average over heart rate estimates from a short-time fourier transform (STFT). Temko et al. [100] refine the heart rate estimate with a phase vocoder that analyses the change in phase between consecutive windows.

For postprocessing, most approaches use techniques to smooth the estimated heart rate trajectory and remove incorrect estimates. Since for the IEEE Signal Processing Cup 2015, the participants were asked to stay still at the beginning of the recording; the first heart rate estimates can be assumed to be correct. TROIKA [113] implements a spectral peak tracking algorithm that tracks the frequency peak corresponding to the heart rate and its seconds harmonic over multiple windows and consecutive peaks that are too far apart. Lakshminarasimha et al. [69] improve the dependency on the initial estimates by initializing the spectral peak tracking algorithm multiple times. Chung et al. [17] improve the estimation accuracy using a finite state machine model with four states (stable, recovery, alert, and uncertainty). The model uses two metrics (crest factor and heart rate change) to estimate the current state, giving more or less importance to the current heart rate estimate. More details about different postprocessing approaches can be found in Section 2.8.

Ballistocardiography-based Heart Rate Estimation

Development of heart rate estimation from BCG is mainly driven by bed-based BCG systems. These incorporate high-resolution single-axis sensors at multiple positions [64, 27] and employ different frequency-based heart rate extraction algorithms, heartbeat waveform analysis, and signal envelope-based analysis [14]. A more ubiquitous approach is to measure the BCG signal from a wrist-worn accelerometer. Especially in real-life settings, the wrist-worn BCG signal is heavily affected by motion artifacts [45] and should therefore be treated carefully. Hernandez et al. [44, 46, 45] use a 4th order Butterworth bandpass filter to first extract frequencies between 4 Hz and 11 Hz. They then combine all axes by computing their magnitude and apply another Butterworth filter of 2nd order, with cut-off frequencies from 0.66 Hz to 2.5 Hz. The resulting signal should represent a clean BCG signal and can be further processed by extracting the heart rate from the FFT spectrum. With this approach, Hernandez et al. reach a MAE of 2.26 bpm on a private dataset. Steffensen et al. [94] decompose the BCG signal using empirical mode decomposition (EEMD) into its components. Zhao et al. [114] evaluate different approaches to extract the BCG signal, namely singular spectrum analysis (SSA), independent component analysis (ICA), variable mode decomposition (VMD), and wavelet synchrosqueezed transform (WSST). For the SSA, they select the BCG components from the X-axis by only taking those components that do not correlate with components from the other axes. This step is followed by a Kalman smoothing filter, achieving an MAE of 3.7 bpm on a private dataset. Zschocke et al. [116] estimate the peak by applying an FFT-based bandpass filter (5 Hz - 11 Hz), followed by a Hilbert transform, and a peak detection algorithm with constraints on the minimum height and the minimum distance between two peaks. Haescher et al. [39] use a high-pass filter on the magnitude of all axes. They compute the heart rate from the squared signal in the FFT spectrum. For

Aspect	Photoplethysmogram (PPG)	Ballistocardiography (BCG)
Measurement Principle	Measures changes in blood volume in tissue microvasculature by detecting light absorption changes.	Measures the body's reaction to the heart's ejection of blood by detecting subtle body movements.
Signal Characteristics	Consists of periodic variations corresponding to heartbeat with pulsatile waveform patterns.	Represents whole-body motion caused by the heart's mechanical activity, with slower frequency compared to PPG.
Applications	Used for pulse rate monitoring, oxygen saturation measurement, and assessment of peripheral vascular conditions.	Used for pulse rate monitoring, monitoring of cardiac function, detecting abnormal heart rhythms, and assessing cardiac output. Can also be used for sleep monitoring.
Sensitivity	More sensitive to peripheral perfusion changes and motion artifacts.	Sensitive to motion artifacts and affected by environmental factors like bed vibrations.
Complexity	Generally simpler to implement and less intrusive.	Can be recorded with any accelerometer, may require specialized equipment and controlled measurement conditions
Spectrum	More sparse with peak and harmonics at heart rate frequency	Generally more noisy, heart rate cannot always be found
Dimensionality	Generally single dimension, multiple wavelengths can be represented with different channels	Mostly three channels for three axes, sometimes single channel aligned with the direction of strongest force

Table 1: Differences between PPG and BCG signals

a summary of signal processing-based approaches for BCG heart rate estimation from the wrist-worn accelerometer, see Table 2.

Deep Learning Approaches for Heart Rate Estimation

Since literature about BCG-related deep learning heart rate estimation is limited, this section will also review approaches from the PPG-domain, where more work has been done recently. Most of the deep learning approaches to heart rate estimation from PPG signals either aim to extract the heart rate directly from the input signal [4, 5, 78, 79, 88] or to extract a cleaned signal with distinct peaks at every heartbeat [12, 84]. Biswas et al. [5] created *CorNET*, a CNN-LSTM network that outperforms many classical approaches on the IEEE Signal Processing Cup Test Set. Similar networks have been implemented with model ensembles [79, 88], as classification [16], or with special attention to memory and energy constraints [7, 8, 80]. Whereas most networks operate in time-domain, Reiss et al. [79], and Song et al. [89] chose to input an FFT-generated signal into their networks. Sarkar et al. [84] constructed a GAN with a cyclic loss and dual discriminator to synthesize a clean ECG signal from the PPG signal. Similarly, Chang et al. [12] construct a clean PPG signal from the corresponding ECG signal and train the network to denoise the PPG signal. Using a hidden Markov model (HMM), Bieri et al. [4] use the output of a dual time-and-frequency network to generate an uncertainty estimate of the predicted heart rate. The inference is done online via belief propagation or offline using the Viterbi algorithm. Moreover, the heart rate is classified into 64 bins instead of doing a regression. Ray et al. [78] model both aleatoric and epistemic uncertainty using Monte Carlo dropout and a negative likelihood loss function. An overview of all approaches can be found in the first half of Table 3 with their reported results.

Most BCG-related deep learning heart rate estimation algorithms aim at extracting the heart rate from a stationary device such as a bed or a chair. At the time of this thesis, no approach was known that aimed at extracting the heart rate from wrist-worn BCG. The found approaches either directly extract the heart rate as a regression problem [86, 111] or try to find the location of the BCG J-peak in the input signal [9, 63, 65]. For direct heart rate estimation, Schubert et al. [86] constructed a CNN-GRU network, evaluating different network hyperparameters and interpreting the model's output using saliency maps. Zhang et al. [111] used a Transformer CNN with a pyramid input to transform a 10-channel BCG signal into a heart rate estimate. Concerning J peak detection, Lu et al. [63] first trained a CNN with a single feed-forward hidden layer and then replaced this layer with an ELM layer. Cathelain et al. [9] and Mai et al. [65] implemented a U-Net architecture to give a probability estimate of the J-peak being at a specific location in the signal. These estimates were then further processed by an adaptive J-peak detection to extract physiologically plausible peaks. See the second half of Table 3 for a summary of BCG-related deep learning approaches.

Self-Supervised Learning

Self-supervised learning is a form of semi-supervised learning that defines domain-related tasks for pre-training the model. It emerged from fields with large amounts of unlabelled data such as computer vision and natural language processing [42]. The idea is for the model to extract meaningful features from the pretext task and, therefore, benefit downstream tasks. A supervisory signal is generated from the unlabelled dataset to train a model. Then, the model's weights are reused on the supervision task, using a form of transfer learning.

Name	Algorithm	Signal	Pre-processing	Results
Zhang [113] *Troika*	Signal Decomposition using SSA; sparse signal reconstruction for high-resolution spectrum estimation with FOCUSS; spectral peak tracking using harmonics	PPG, ACC	Butterworth (2n, 0.4 Hz - 5 Hz), downsampled to 25 Hz	2.42 bpm MAE on IEEE SPC Train
Zhang [112] *Joss*	Signal Decomposition using SSA; multiple measurement sparse signal reconstruction	PPG, ACC	Butterworth (2n, 0.4 Hz - 5 Hz), downsampled to 25 Hz	1.28 bpm MAE on IEEE SPC Train
Lakshminarasimha [69] *MISPT*	MA removal by adaptive noise cancellation, Spectral peak tracking with multiple initializations as opposed to [113]	PPG		1.28 bpm MAE on IEEE SPC Train
Temko [100] *WFPV*	Wiener filter to remove MA in PPG using noise signatures from acceleration; phase vocoder for high-resolution frequency estimation	PPG	Butterworth (4n, 0.4 Hz - 4 Hz), z-score normalization, averaged, downsampled to 25 Hz	1.02 bpm MAE on IEEE SPC Train
Salehizadeh [82] *SpaMA*	Spectral Analysis, Comparison of acceleration spectrum with PPG spectrum	PPG, ACC	Bandpass filter (0.5 Hz - 3 Hz), downsampled to 31.25 Hz	3.36 bpm MAE on IEEE SPC Train
Chung [17]	MA removal with Wiener Filter using accelerometer spectrum. HR estimation with Finite State Machine, using two metrics (Crest Factor, HR Change), four states (stable state, recovery state, alert state, uncertainty state)	PPG, ACC	Butterworth (4th, 0.4 Hz - 4 Hz), z-score normalization, averaging, downsampled to 25 Hz	0.79 bpm MAE on IEEE SPC Train
Boloursaz [67]	MA removal by iteratively removing ACC components (SVA) with frequencies in (0.4 Hz,5 Hz), Spectrum Estimation using sparse signal reconstruction with IMAT, spectral peak tracking using harmonics (similar to TROIKA)	PPG, ACC	Bandpass filter (40-100 bpm), downsampled to 25 Hz	1.02 bpm MAE on IEEE SPC Train
Zhao [115] *SFST*	No MA Removal, Baseline drift filtering, STFT and spectral analysis (detection, verification, prediction)	PPG	zero-phase forward and reverse IIR filter	1.06 bpm MAE on IEEE SPC Train
Hernandez [46, 44, 45] *BioInsights*	Butterworth Bandpass (4nd, 10 Hz - 14 Hz), L2-norm of axes, Frequency peak detection using FFT	BCG, Gyroscope	z-score normalization, detrending	2.26 bpm MAE on private dataset
Steffensen [94]	Signal segmentation using tonometer, BCG wave averaging, head-to-foot axis selection, EEMD decomposition	BCG, Tonometry		
Zhao [114]	SSA decomposition, compute principal component of (x,y), (x,z) axis, component selection if the principal component is smaller than 0.6, HR estimation through peak detection	BCG		3.7 bpm MAE on private dataset
Zschocke [116]	Bandpass Filter (FFT, 5 Hz - 14 Hz), Hilbert Transform, Peak detection (threshold, minimum distance 0.5 s), Axis selection (sanity check, autocorrelation max)	BCG	Setting acceleration values to zero if MAD exceeds the threshold, subtraction of mean per second	0.6 % error on private dataset
Haescher [39]	Magnitude computation, High-pass filter, square signal, FFT heart rate extraction	BCG		1.63 % error on private dataset

Table 2: Summary of signal processing-based algorithms and techniques for PPG and BCG-based heart rate estimation.

Name	Algorithm	Signal	Pre-processing	Results
Biswas [5] CorNET	CNN-LSTM network	PPG	8 s window, 2 s sliding, Butterworth Band Pass (2n, 0.1 Hz-18 Hz), z-normalization	1.99 bpm MAE on IEEE SPC Train
Chang [12] *DeepHeart*	Generative CNN network for generation of clean PPG signal	PPG	Bandpass (0.4 – 5 Hz), 8 s window, 0.25 s slide	1.61 bpm MAE on IEEE SPC Train
Reiss [79] *DeepPPG*	CNN on spectrogram	PPG, ACC	8 s window, 2 s slide, FFT spectrum computation, 3 windows together as input	4 bpm MAE on IEEE SPC Train
Bieri [4] *BeliefPPG*	Multimodal Attention U-Net/CorNET with Attention, probabilistic postprocessing	PPG	Z-normalize, band-pass (0.1-18 Hz), resample to 64 Hz, 8s window, 2s slide	1.47 bpm MAE on IEEE SPC Train
Ray [78] *DeepPulse*	CNN-LSTM network with uncertainty head and MCdropout	PPG, ACC	Bandpass (2n, 0.5 Hz - 4.5 Hz), z-normalize, 8s window, 2s slide	2.76 bpm MAE on IEEE SPC Train
Shyam [88] *PPGNet*	CNN-LSTM on time-domain	PPG	Bandpass (2n, 0.5 Hz - 5 Hz), 8 s window, 2 s slide	3.36 bpm MAE on IEEE SPC Train
Song [89] *Nas-PPG*	CNN-LSTM on frequency domain	PPG	Bandpass (4n, 0.4 Hz - 4 Hz), 8 s window, 2 s slide, FFT (256 points, 0.6 Hz - 3.6 Hz)	0.82 bpm MAE on IEEE SPC Train
Sarkar [84] *CardioGAN*	Two Attention-UNets for generation of ECG from PPG signal, Discriminators in time and frequency domain, Adversarial Loss, Cyclic Consistency Loss	PPG (ECG)	Bandpass (1 Hz - 8 Hz), z-score normalization, 4s window, 3.6 s slide, min-max normalization	0.7 bpm - 8.6 bpm MAE on different datasets
Burrello [8, 7] *Q-PPG*	TCN network with optimized number of feature maps and dilation per layer, quantization	PPG, ACC	Comparison with mean HR, clipping	3.61 bpm MAE on PPG-Dalia
Chung [16]	CNN-LSTM, Power Spectrum Input (0.3 Hz - 3.3 Hz), Classification into discrete probabilities	PPG, ACC	Bandpass (4n, 0.4 Hz - 4 Hz), 8 s window, 2 s slide, FFT (256 points, 0.6 Hz - 3.6 Hz)	0.67 bpm MAE on IEEE SPC Train
Zhang [111] *HRCTPNet*	Transformer CNN with pyramid input and ECG noise	BCG (10 channels) under mattress	Extract clean signal, 10 s window, 10 s slide, augmented with ECG noise	0.93 bpm MAE on private dataset
Cathelain [9]	U-Net for J peak identification, adaptive J peak detection	BCG (2 channels) under mattress		5.7 bpm MAE on private dataset
Schubert [86]	CNN-GRU	1-channel BCG foam pad		2.07 bpm MAE on private dataset
Lu [63]	CNN-Extreme Learning Machine for peak detection. First train CNN, then replace the head with ELM and train ELM	BCG (1 channel) on chair	Denoising filter, manual verification	1.36 % classification error rate on private dataset
Mai [65]	U-Net, Bi-LSTM for J peak detection, adaptive J peak detection	BCG (1-channel) under pillow	Bandpass (1 Hz - 10 Hz), SNR-metric	97.59 % precision on private dataset

Table 3: Summary of deep learning-based algorithms and techniques for PPG and BCG-based heart rate estimation.

Generally, self-supervised learning (SSL) can be separated into three categories [61]:

1. *Generative*: The model is trained to reconstruct a distorted version of the original signal. Typically, this is achieved with an encoder-decoder structure like auto-encoders.

2. *Contrastive*: The model is trained to learn a latent representation of the data, such that samples that are close in reality are also close in the latent space. Common architectures can be diverse, but all have an encoder-like structure.

3. *Generative-Contrastive* (also called adversarial representation learning): Combining the two approaches above, the model is trained to reconstruct a sample, which is then fed into a discriminator. The discriminator is trained to distinguish between the true and the generated sample. Typically, this is achieved with GAN-like structures, but also other models like Autoencoders are used.

Popular contrastive SSL-approaches include SimCLR [13], SimSiam [13], and BYOL [37]. SimClR trains an encoder model by contrasting different augmented views of the same input with other windows from the same batch serving as negative samples. BYOL consists of two networks receiving different augmented versions of the same input window. Minimizing the MSE between the networks' outputs, only the weights of one network are updated. In contrast, the other network is updated via a slow-moving average of the first network. This eliminates the need for negative pairs. Building upon this structure, SimSiam also uses Siamese networks but uses a stop-grad operation instead of the moving average updates to prevent the network from collapsing.

Transfer learning [103] is used to transfer the learned information to the target task. This usually involves taking the pre-trained encoder and further processing its output with a second network, such as a classification head. To prevent catastrophic forgetting [34], the encoder's layers are frozen, and only the second network is trained in a supervised fashion.

Self-Supervised Learning in Wearable Data

Although self-supervised learning has mostly been popular in natural language processing and computer vision, it has also been used in fields that include wearable data such as human activity recognition (HAR) [42]. Unfortunately, self-supervised learning has not yet been applied to heart rate estimation tasks from PPG or BCG. However, previous work has shown that SSL helps to extract features from the input modality independent of the task and, therefore, helps generalization on the target domain [49]. Hence, applying techniques that proved to work on accelerometer data like human activity recognition also has prospects of performing well on BCG data. Looking at the state-of-the-start of SSL for human activity recognition is thus the goal of this section.

In the context of HAR, SSL has been explored to extract informative features from accelerometer data collected from mobile phones, wristbands, and ankle bands [81]. Initial studies by Saeed et al. [81] applied eight transformations to accelerometer data in a multi-task setting, aiming to capture core signal characteristics such as high-level semantics, sensor behavior under different placements, temporal shifting, varying amplitudes, and robustness against sensor noise. These transformations included noise addition, scaling, rotation, negation, flipping, permutation, time warping, and channel shuffling. Through this training scheme, the model learns to independently predict whether each transformation is applied, facilitating effective representation learning. Post-training analysis involves evaluating the model's performance on activity recognition tasks along with feature space explorations [81].

Moreover, Yuan et al. [18] applied multi-task-learning on the UK Biobank dataset with more than 700,000 person-days of wearing [109], finding that arrow-of-time and permutation augmentations benefit the downstream task performance the most. They also showed that the learned features generalize well across datasets, tasks, devices, and populations and that a whole-network fine-tuning approach increases performance the most.

Additionally, transformer encoders have been employed for self-supervision in HAR, reconstructing sensory data at randomly masked timesteps [40]. This approach captures local temporal dependencies in the data by reconstructing information only at the masked timesteps. The trained encoder layers are utilized for downstream tasks such as activity recognition.

Furthermore, SelfHAR combines teacher-student self-training and multi-task self-supervision to enhance diversity in training data, resulting in more generalizable features [98]. This method leverages knowledge distillation to train a teacher model on labeled data, which is then used to pseudo-label a large-scale unlabeled dataset. The combined datasets are used for pre-training with multi-task self-supervision, and the learned encoder weights are frozen for subsequent activity recognition tasks. SelfHAR experiments extend to using wrist-based data from the *Fenland* [33] dataset and evaluating performance across different sensor locations, along with analyzing the effect of dataset composition [98].

Contrastive Predictive Coding (CPC) has also been applied to HAR, where windows of accelerometer and gyroscope data are encoded to predict future timesteps [41]. CPC captures local variations and long-term dependencies in the data, resulting in strong representation learning.

Another contrastive learning approach was proposed by Tang et al. [99], resulting in an adaption of the SimCLR [13] framework. By investigating different augmentations and their effect on downstream task HAR performance, they showed that rotation together with a fine-tuning approach achieves the best results.

In a large study across different datasets, sensor locations, and SSL-frameworks, Haresamudram et al. [42] evaluated the robustness to differing source and target conditions, the influence of dataset characteristics, and feature space characteristics. The results indicate that SSL generalized well for different downstream task activities and sensor locations, especially with complex classifiers such as MLP. Another finding is that the sampling rate between two datasets should be identical for good performance and that the SSL-model learns

meaningful features for downstream tasks, even with limited but diverse data available.

Likewise, Qian et al. [77] evaluate the effect of different SSL-frameworks, model architectures, and augmentations on the downstream task performance. They find that CNN and LSTM networks achieve the best results, while different augmentations are optimal for different SSL-frameworks. However, SimCLR and BYOL outperform other models on most downstream modalities, and NNCLR performs best under mild domain shifts.

Many datasets in the domain of mobile health are multimodal (i.e., ECG, PPG, BCG, different sensor positions). Jain et al. [51] use this fact by forcing positive samples to be close together and far from the negative samples in the latent space, similar to SimCLR. However, instead of augmenting the samples to generate positive samples, they use time-synchronized windows from other sensor locations to generate positive and negative (time-asynchronized) windows from other sensor locations to generate negative samples.

In a similar fashion, Spathis et al. [90] used the heart rate annotations to pre-train a model that predicts the future heart rate based on acceleration values using a customized quantile loss function. The model is then transferred to predict certain metadata such as Body-Mass-Index (BMI), sex, gender, and age. The model learns physiologically meaningful representations on a user level, achieving high classification accuracies on personalized downstream tasks.

Uncertainty Estimation

Uncertainty estimation is a critical element of a well-calibrated deep learning model. A goal of uncertainty estimation is to estimate the predictive uncertainty of the model. That is the uncertainty of a prediction y^* given a sample x^*, independent of the data D that the model is trained on,

$$p(y^*|x^*) = \int_D p(y^*|x^*, D) \tag{1}$$

Usually, the distribution from Equation 1 is unknown and can only be estimated given a sampled dataset $D = \{x_i, y_i\}_{i=1}^n$, which eliminates the integral in Equation 1.

$$p(y^*|x^*, D) = p(y^*|x^*, x_{1:n}, y_{1:n}) \tag{2}$$

Further simplifications can be made by taking the maximum over the predictive distribution, which yields the maximum a posteriori (MAP) estimate, a point estimate of y^*.

$$y^*_{MAP} = \arg\max_{y^*} p(y^*|x^*, D) \tag{3}$$

Going a step back, uncertainty estimation would like to model the distribution of y^* given a sample x^* and the data D (Equation 2). In machine learning models, such as deep neural networks, the function that maps x^* to y^* is parameterized by parameter space θ, also commonly referred to as the model's weights. The predictive uncertainty can then be obtained by applying Bayes' theorem and marginalizing out θ.

$$p(y^*|x^*, D) = \int \underbrace{p(y^*|x^*, \theta)}_{\text{epistemic}} \underbrace{p(\theta|D)}_{\text{aleatoric}} d\theta \tag{4}$$

Generally, two types of uncertainty can be modeled: Aleatoric uncertainty and epistemic uncertainty [52]. Aleatoric uncertainty, also called data uncertainty, captures the noise inherent to the observations and cannot be reduced. Epistemic uncertainty, also known as model uncertainty, can be understood as uncertainty in the model parameters. It can, therefore, be reduced by increasing the amount of training data.

Since the right-hand side of the equation 4 is generally computationally intractable, different ways of approximating it have been introduced, such as Monte Carlo dropout, BNNs, deep ensembles [25], and evidential neural networks [2]. An extensive overview of different models can be found in Gawlikowski et al. [32]. Generally, uncertainty models can be separated into three classes: Single deterministic, Bayesian, and snsemble.

The model's parameters θ can be estimated using the MAP of the aleatoric part in Equation 4.

$$\begin{aligned}
\hat{\theta}_{MAP} &= \arg\max_{\theta} p(\theta|D) \\
&= \arg\max_{\theta} p(\theta|x_{1:n}, y_{1:n}) \\
&= \arg\max_{\theta} p(y_{1:n}|x_{1:n}, \theta) \cdot p(\theta) \\
&= \arg\max_{\theta} \underbrace{\log p(\theta)}_{\text{Prior}} + \underbrace{\sum_{i=1}^{n} \log p(y_i|x_i, \theta)}_{\text{Likelihood}}
\end{aligned} \tag{5}$$

Modeling the prior as a Gaussian distribution yields the L2-norm, and modeling as a Laplace distribution yields the L1-norm of the model's parameters θ. If the prior is removed, θ will be determined as the maximum likelihood estimation (MLE):

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \sum_{i=1}^{n} \log p(y_i|x_i, \theta) \tag{6}$$

In non-Bayesian deep learning models, the network outputs deterministic point estimates of y given x with non-linear functions. It is usually trained by maximizing equation 5 or 6 with a form of gradient descent. The prior from Equation 5 then acts as a regularizer on the model's parameters.

Single Deterministic Models

Single deterministic models model the epistemic uncertainty by directly predicting the distribution $p(y^*|x^*, \theta)$ in a single forward pass. An example is a general classification task trained with the cross-entropy loss. The softmax output can be interpreted as probabilities of the sample belonging to either

of the classes. However, it has been shown that these estimates tend to be overconfident [87]. Simple measures can help calibrate the network and improve its uncertainty estimate by, for example, multiplying the network's output before the softmax layer with a learnable parameter [38], which is optimized on the validation set. For regression tasks, the network usually outputs a parameterized distribution like a Gaussian by predicting the parameters of the distribution using two regression heads.

Modelling the output probabilities $p(y_i|x_i,\theta)$ of each i.i.d. sample as Gaussian $\mathcal{N}(y_i|\mu(x_i),\sigma^2(x_i))$, with σ and μ parameterized by θ, results in the negative log likelihood function

$$\begin{aligned} L_{NLL}(\theta) &= \sum_i -\log p(y_i|\mu(x_i),\sigma^2(x_i)) \\ &= \sum_i \frac{1}{2}\log(2\pi\sigma(x_i)^2) + \frac{(y_i-\mu(x_i))^2}{2\sigma(x_i)^2} \end{aligned} \tag{7}$$

Minimizing this function gives the MLE of the model parameters. The network learns to predict the uncertainty in the data, specifically the aleatoric uncertainty. Amini et al. [2] extend this model by placing Normal Inverse-Gamma distribution prior to the mean and variance and extracting an estimate for both aleatoric and epistemic uncertainty. To achieve better robustness against outliers, Nair et al. [70] modelled the output labels with a Laplace distribution. In summary, single deterministic models stand out as they are computationally efficient since they only need one forward pass. However, they heavily rely on the underlying network architecture, training procedure, and other hyperparameters.

Bayesian Neural Networks

In BNNs, the goal is to model the uncertainty in the network's parameters θ instead of using the MAP estimate from equation 5. This is done by imposing a prior on the parameters, for example, a Gaussian:

$$p(\theta) = \mathcal{N}(0,\sigma_p) \tag{8}$$

This makes the integral in 4 intractable, which is why there are different techniques for approximating the posterior of θ. Applying Bayes to the epistemic uncertainty yields:

$$p(\theta|x_{1:n},y_{1:n}) \propto p(y_{1:n}|x_{1:n},\theta)\cdot p(\theta) \tag{9}$$

Variational Inference, also called *Bayes by Backprop* [6] approximates the posterior of θ with a variational family of parametric distributions $q(\theta)$. The respective parameters are learnt minimizing the Kullback-Leibler (KL) divergence between the variational $q(\theta)$ and the posterior $p(\theta|x_{1:n},y_{1:n})$. Since the latter is intractable, an alternative approach is to minimize the evidence lower bound (ELBO).

For inference, the distribution of $p(y^*|x^*,D)$ can be estimated by sampling multiple predictions from the variational $q(\theta)$.

$$p(y^*|x^*,D) \approx \frac{1}{m}\sum_{i=1}^{m} p(y^*|x^*,\theta^{(i)}) \quad \text{with } \theta^{(i)} \sim q \tag{10}$$

Other approaches to estimate the posterior of θ are Markov Chain Monte Carlo (MCMC), Laplace approximations [32] and Monte Carlo dropout [30]. Monte Carlo dropout has especially seen a lot of attention for its simplicity and applicability. dropout is a regularization technique that randomly disregards specific nodes in a layer at training time. Gal et al. [30] formulate the layers as Bernoulli distributed random variables and apply dropout at testing time. By sampling multiple predictions, the predictive uncertainty can be estimated.

BNNs are a valuable family of models to estimate the epistemic uncertainty. They are not sensitive to the model's hyperparameters. However, choosing a good prior is a crucial task, and BNN are more computationally expensive at training and inference than single deterministic models.

Ensemble Models

Ensemble models, also called probabilistic ensembles, combine the prediction of multiple different deterministic models at inference. This originates from the idea that a group of decision-makers tends to make a better decision than a single decision-maker. It has been applied in many domains of machine learning, such as Random Forests and Gradient Boost, usually as a measure to improve performance. However, it can also be used to estimate uncertainty, for example, if the group of predictors disagrees. Several factors and hyperparameters can be varied to maximize variance between predictors. These include:

- Training (Weight Initialization, Loss Function, Local Optima [48])
- Model (Architecture [47])
- Training Data (bagging, boosting [62], augmentations [71], shuffling)

An estimate of the predictive uncertainty can then be found by obtaining m MAP estimates of the parameters $\theta^{(i)}$ under different hyperparameters:

$$p(y^*|x^*,D) \approx \frac{1}{m}\sum_{i=1}^{m} p(y^*|x^*,\theta^{(i)}) \tag{11}$$

An advantage of ensemble models is that they can model different optima [25]. Training multiple predictors can result in different local optima, leading to multi-mode evaluation. BNNs and single deterministic models, on the other hand, only capture a single optimum and its variance and hence perform single-mode evaluation.

A downside of ensemble models is that they are computationally expensive in both training and inference. Since every predictor is trained independently, the computational cost grows

with the amount of predictors that are being sampled. Solutions that have been proposed include weight sharing, pruning, and ensemble distillation [32].

Uncertainty Calibration Evaluation

Uncertainty calibration refers to assessing and adjusting the level of uncertainty associated with a measurement or prediction to make it more accurate and reliable [38]. In a classification setting, the probability output of a classifier should represent the true probability of correctness [32, 57].

$$\forall p \in [0,1] : \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{y_{i,k} \cdot \mathbb{I}\{f_\theta(x_i)_k = p\}}{\mathbb{I}\{f_\theta(x_i)_k = p\}} \xrightarrow[N\to\infty]{} p \qquad (12)$$

Here $\mathbb{I}(\cdot)$ is the indicator function that is either 1 if the condition is true or 0 if it is false, and $y_{i,k}$ is the k-th entry in the one-hot encoded ground truth vector of a training sample (x_i, y_i)

For a regression problem, the predicted confidence intervals should match the confidence intervals computed from the data [32, 57].

$$\forall p \in [0,1] : \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left\{y_i \in \text{conf}_p(f_\theta(x_i))\right\} \xrightarrow[N\to\infty]{} p \quad (13)$$

Here, conf_p is the confidence interval that covers p percent of a distribution. It is estimated from the uncertainty output. This means that for every probability, the share of true values that fall into the confidence interval defined by the probability and the output distribution, should converge to the probability itself.

The confidence can be plotted against the accuracy for different quantiles p to evaluate the calibration error. This is called a reliability diagram, and it gives an estimate in which area of confidence the model's accuracy deviates. For regression, the confidence can be computed directly from the quantile p. For classification, the confidence scores are usually grouped into bins and evaluated for each bin. If the confidence is higher than the accuracy, the model is called overconfident for that quantile. If the confidence is lower than the accuracy, the model is called underconfident for that quantile. A standard metric to summarize reliability diagrams is the expected calibration error (ECE) [38, 72]. For classification it can be written as in Equation 14:

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)| \qquad (14)$$

Here, $\text{conf}(B_m)$ represents the average confidence of all predictions that fall into bin m. $\text{acc}(B_m)$ represents the share of correct predictions out of all the predictions falling into the same bin m. Two characteristics differ between the expected calibration error (ECE) and reliability diagram. For ECE, the bins are weighted, which means that bins with more predictions are weighted higher. Moreover, instead of taking the average confidence per bin, the average confidence of all predictions per bin is taken. In a reliability diagram, all samples that fall into bin m are classified as having a confidence that is computed from the position and width of the bin, but not from the distribution of the samples inside the bin.

Postprocessing of Heart Rate Predictions

As mentioned in Section 2.4 and Section 2.5, there are a few approaches that postprocess the heart rate estimates using probabilistic and deterministic approaches. These approaches mostly exploit the physiological background of the heart rate being time-dependent. They can, therefore, correct wrong heart rate estimates caused by noise and motion artefacts. Chang et al. [11] model the heart rate estimate as a finite state machine, having four states that determine the credibility of the current heart rate estimate. *Troika* [113] and its successors implement postprocessing by searching for the heart rate only in the close surroundings of the previous heart rate estimate. Stegle et al. [95] employ a two-stage postprocessing, where they first cluster the sample into different noise classes and then infer the latent heart rate using a Gaussian process with the estimated noise class. Zhao et al. [115] use Kalman smoothing to postprocess heart rate estimates. Similarly, Temko et al. [100] employ a Viterbi decoding postprocessing step. Bieri et al. [4] model the system as a hidden Markov model (HMM), with the heart rate being the hidden state, and the accelerometer signal being the observation. The transition probability is learned from the data and modeled as a Laplace distribution. For the heart rate function, a probabilistic deep learning model is used; in this case a simple classification model with labels modeled as Gaussian. The model outputs the state probability as discrete values over all possible heart rates and is used as input to the postprocessing model.

SETUP

The following section describes the conception and implementation of different approaches to estimate the heart rate from wrist-worn BCG. We introduce three datasets to train and evaluate the model. The first group of models are classical signal processing approaches, inspired by previous works in Table 2. The second group of models considers deep learning models, as in Table 3. The primary attention will lay on implementing and evaluating different components of deep learning models.

Datasets

Capture24

Capture24, a large dataset for human activity recognition [10], includes recordings from 151 participants over approximately one day, totaling around 4,000 hours of data. The data comes from a single Axivity AX3 wrist-worn activity tracker, sampled at 100Hz. Annotations were made using Vicon Autograph wearable cameras and Whitehall II sleep diaries, resulting in over 2,500 hours of roughly labeled data. These annotations detail over 200 specific activities or six broader categories like sleep, sitting or standing, mixed activities, walking, vehicle use, and bicycling. The authors also provide mappings between detailed labels and broader categories for sleep tracking and activity levels. Unless otherwise stated, only windows labeled as "sleep" are used for this work.

Apple Watch Dataset

The Apple Watch Dataset [102] includes acceleration and PPG-based heart rate recorded from the Apple watch. It was conducted at the University of Michigan and includes 7-14 days of recording for each of the 39 participants. The age range is between 19 and 55, with a mean age of 29.4, and the gender distribution is balanced. Each participant spent one night in a sleep lab, where sleep stages were recorded, and the Apple Watch recorded heart rate and acceleration. Each acceleration window was labeled with the mean heart rate for that window, given by the Apple watch. Windows with no or constant heart rate were excluded. The acceleration was recorded at 50 Hz and up-sampled to 100 Hz, with a maximum amplitude of 2 g.

In-House Dataset

We recorded an in-house dataset, consisting of 23 participants, of whom 15 were male and 8 female. The minimum age is 21 years, and the maximum is 36 years, with a mean age of 26.9 years. All participants were asked to wear an Axivity AX3 wrist-worn activity tracker and a Movisense ECG chest-band for one night. The Movisense algorithm extracted the location of the ECG R-peaks and a quality metric for each R-peak. From these R-peaks, the instantaneous heart rate for each window was computed as seen in Equation 15, using only valid RR intervals. For each window, the R-peaks from the surrounding 5 seconds were also included for the HR estimation to get a more stable heart rate trajectory. All windows with less than 4 valid RR peaks were excluded. The acceleration signal was recorded at 100 Hz, with a maximum amplitude of 8g, and the ECG-signal was recorded at 1024 Hz.

$$HR[t] = \frac{60\frac{s}{min}}{\frac{1}{N}\sum_{k \in window_{t\pm 5s}} RR_k} \tag{15}$$

M2Sleep Dataset

The M2Sleep dataset was recorded from 16 participants, of which 5 were female and 11 male, between 19 and 35 years old [31]. The participants were instructed to wear an Empatica E4 wristband each night plus 4 hours before going to sleep and after waking up for 30 days. The recorded data includes PPG-derived heart rate, 3-axis acceleration, and electrodermal activity. In addition, the participants were asked to report their sleeping times and perceived sleep quality. We up-sampled the acceleration signal from 32 Hz to 100 Hz, with values ranging between -2 g and 2 g. The acceleration value resolution of the Empatica E4 is 8 bit within the acceleration range. This corresponds to a resolution of 0.0156 g with ±4 g and 0.0078 within ±2 g.

Preprocessing

We resampled all datasets to 100 Hz and preprocessed by applying a Butterworth 4th order bandpass filter with cutoff frequencies at 0.1 Hz and 18 Hz. Next, we split all traces into 10-second windows with a step size of 8 seconds. To further eliminate baseline drift and prepare the samples as input into the neural network, we z-score normalized each window and channel using their means $\mu_{i,ch}$ and variances $\sigma^2_{i,ch}$:

$$X_{i,ch}[t] = \frac{X_{i,ch}[t] - \mu_{i,ch}}{\sigma_{i,ch}}$$

After preprocessing, the Apple Watch dataset contained 85,442 samples, the in-house dataset contained 64,328 samples, and the M2Sleep dataset contained 1,509,125 samples.

We split the samples into train and test sets, allocating 20 % of the original set for testing. We then divided the training set into training and validation sets using 5-fold cross-validation. We performed all splits in a subject-wise manner, ensuring no subject appeared in more than one partition for any split. The splits remained fixed for the remainder of the experiments. The test set was kept constant across all folds and evaluated only at testing time to prevent any leakage of test data into the training data.

Evaluation

A metric was needed to assess the performance of different models on a given test dataset. Following previous publications, we used mean absolute error (MAE), as it is a common metric (see Table 3) to assess the performance of a regression model. It compares the predicted values y_{pred} with the true values y_{true} by taking the average over the absolute error. It is given in the unit of the estimated value and is scale-dependent:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_{true,i} - y_{pred,i}| \tag{16}$$

However, the mean absolute error (MAE) does not always accurately reflect how well a predictor follows the underlying trajectory. For example, if we use the median heart rate from the training set to estimate the heart rates in the validation and test sets of our in-house dataset, the MAE is 6.06 bpm. To address this limitation, we also used Pearson's correlation coefficient to measure the relationship between predicted values and true values as an additional metric. The Pearson correlation coefficient is computed as follows:

$$Corr = \frac{\sum_{i=1}^{N}(y_{true,i} - \overline{y_{true}})(y_{pred,i} - \overline{y_{pred}})}{\sqrt{\sum_{i=1}^{N}(y_{true,i} - \overline{y_{true}})^2}\sqrt{\sum_{i=1}^{N}(y_{pred,i} - \overline{y_{pred}})^2}} \tag{17}$$

Pearson's correlation coefficient ranges from -1 to 1, where 1 means that the predictor perfectly follows the ground truth value up to a scale factor and offset. However, it has been criticized [60] for being biased and dependent on the underlying dataset and should therefore be used carefully and only in combination with the MAE.

Additionally, we performed a qualitative analysis of the results using Bland-Altman plots and showing the heart rate trajectory with the corresponding predictions.

Signal Processing Approaches

As a first step, we implemented different signal-processing-based approaches from Table 2 to estimate the heart rate from

the three datasets in Sections 3.1.2, 3.1.4, 3.1.3. They all contain wrist-worn accelerometer signals. Specifically, we implemented and adapted the following approaches to the BCG-domain:

- ***BioInsights*** [46]: A 4th-order Butterworth bandpass filter with cutoff-frequencies at 4 Hz and 11 Hz was applied on all channels, followed by taking the L2 norm over all channels. A "clean" BCG signal was extracted, applying another 2nd order Butterworth bandpass filter with cutoff-frequencies at 0.5 Hz and 2.0 Hz. In the original approach the heart rate was computed from the spectrogram of the filtered signal. We adapted this by extracting J-peaks using a continuous wavelet approach by Du et al. [19]. Given the J-peaks, we computed the heart rate by taking the inverse of the mean JJ intervals (see Equation 15).

- **Zschocke et al.** [116]: After applying a bandpass filter with cutoff-frequencies at 4 Hz and 14 Hz, a Hilbert transform was applied on each axis and added to the original signal as an imaginary part. From the amplitude of the resulting signal, the axis was chosen, which results in heart rate estimates between 40 bpm and 200 bpm. If this criteria applied to multiple axes, the axis with the highest auto-correlation function between 0.4 s and 1.5 s was chosen. The J-peaks were then extracted using a peak detection algorithm with a minimum distance of 0.5 s. In the original paper, 74.3 % of all recordings in their private dataset had been excluded by applying a threshold on the mean amplitude. We changed the peak detection algorithm to the same approach as in *BioInsights* to guarantee compatibility and increase performance.

- **Steffensen et al.** [94]: First, each axis of the signal was decomposed using a singular spectrum analysis (SSA). From the resulting eigenvalues-eigenvectors pairs, eigenvalues were selected by correlating the X-axis eigenvectors with the other two axes and taking the components whose correlation exceeds a threshold. Then, the L2-norm of all axes was taken, and a 4th-order Butterworth bandpass filter with cutoff frequencies at 0.5 Hz and 2 Hz is applied. The heart rate was extracted by similar means as in *BioInsights*. In the adaptation, we selected the 20 largest eigenvalues for each axis. This was motivated by performance issues on the original implementation.

- ***Troika*** [113]: This approach contains a SSA-decomposition, followed by a grouping step where all components were clustered based on the correlations of their reconstructed signals. Then, specific groups were selected based on their frequency components. Since we adapted this approach from the PPG-domain, we applied the first steps for all three axes of the acceleration signal. We then selected the group whose frequency components in the range of 0.5 Hz and 2 Hz exceeded a threshold to reconstruct a "clean" signal. Subsequently, we merged all three axes using the L2-norm and computed the spectrum from the resulting signal using a FFT. We chose the highest peak with frequencies between 0.5 Hz and 2 Hz to estimate the heart rate. In the original approach, the spectrum was computed using a sparse signal reconstruction (SSR). However, this did not result in effective performance, which we contribute to the BCG-signal being less sparse than the PPG-signal.

Deep Leaning Frameworks

Deep learning serves as a common alternative to classical signal processing. To evaluate the influence of different hyperparameters on model performance, we defined and implemented a baseline model with a set of baseline parameters. For simplicity, we assumed that the effects of different groups of hyperparameters, such as postprocessing and pre-training, were orthogonal, meaning there is no interaction between them. The implementation was based on the library by Qian et al. [77], and we adapted further methods from Bieri et al. [4].

Baseline Model

We began by using the library from Qian et al. [77] as our initial framework to identify a suitable baseline model, assessing various parameters accordingly. We trained each model for 60 epochs and selected the best model based on its performance on the validation set. Finally, we evaluated the parameters on the test set using subject-wise 5-fold cross-validation.

1. **Model Architecture**: The models consist of a feature extractor and a classification head, set to be a single fully connected feed-forward layer. The feature extractor architectures are:

    - **CNN**: The convoluational neural network (CNN) model includes 3 blocks, each with 64 convolutional kernels with a size of 8, a batch normalization layer, a max-pooling layer, and a dropout layer.
    - **CorNET** [5]: This model has 2 convolutional blocks, each with a convolutional layer, followed by a batch-normalization layer, a max-pooling layer, and a dropout layer. The extracted features are fed into two one-directional GRU layers, with 128 units in each layer. In the original paper, an LSTM was used, but the GRU proved to perform better.
    - **GRU**: The gated recurrent unit (GRU) model consist of 2 one-directional GRU layers, with 128 units per layer.
    - **Transformer**: Positional encoding is added to the input. It is then fed into the transformer model, which consists of a linear layer followed by 4 identical blocks. The linear layer converts the input to a 128-dimension embedding space. A token of size 128 is added to the embedded input as the representation vector. Each block has a multi-head self-attention layer, a fully connected feed-forward, and a residual connection.
    - **HRTCPNet** [111]: HRTCPNet is built of 4 convolutional layers with downsampling skip-connections, followed by a Transformer-encoder with 6 layers, and a single final dense layer.

2. **Model Hyperparameters**: After choosing a model, we optimized different model hyperparameters, including the number of layers, kernel size, number of RNN units, and dropout rate. First, we ran a Bayesian optimization on a

single fold to estimate a baseline set of parameters. We then deviate the baseline parameters step-by-step according to the following list:

- **Number of Kernels**: We tried values 16, 32, 64 for the number of kernels per convolutional layer.
- **Kernel Size**: The evaluated kernel sizes for the convolutional layers are 4, 8, 16. The pooling layer kernel sizes stay constant.
- **Number of RNN Units**: The possible values are 64, 128, 196. The number of RNN layers stays constant at 2.
- **Dropout Rate**: Dropout rates are evaluated for 0.1, 0.3, and 0.5.
- **RNN architecture**: Different RNN architectures are evaluated, mainly LSTM, GRU, and bi-directional LSTM and GRU.

3. **Loss Function**: Since heart rate estimation is a regression problem, we evaluated the following loss functions:
   - **MSE**: The mean squared error loss function is most commonly used and is smooth around 0.
   - **MAE**: The mean absolute error loss function is robust against outliers and noise in the dataset [76].
   - **Huber Loss**: The Huber loss function combines the advantages of MSE and MAE with being smooth around 0 but still robust to outliers. We configured the Huber loss with a delta value of 0.1.
4. **Learning Rate**: The learning rate stays constant over the training process and is evaluated for the values between $1 \cdot 10^{-3}$ and $1 \cdot 10^{-5}$
5. **Batch Size**: The batch size is commonly set to 64 in comparable works. However, we evaluate batch sizes up to 1024 to accelerate training and smooth the gradient.

Self-Supervised Learning

Self-supervised learning (SSL) intends to extract meaningful features from the data by leveraging unlabeled datasets. Since the amount of labeled data for this task (see Section 3.1) is limited in terms of variety and amount, self-supervised learning promised to improve performance and robustness. Different SSL-frameworks were implemented and tested. In addition, we tested different pre-training augmentations, the amount of unlabeled data for pre-training, and the amount of labeled data for fine-tuning. We then compared the constructed models to the baseline model from Section 3.6.

The following SSL parameters were evaluated:

1. **SSL Framework**: We compared different, mostly contrastive learning frameworks that have been tested on HAR-tasks. The tested frameworks include BYOL [37], SimSiam [13], SimCLR [13], NNCLR [20], and TS-TCC [21], and Multi-Task pre-training [109]. For each framework, we kept the optimal hyperparameters excluding augmentations from Qian et al. [77], who optimized them on SHAR, a HAR dataset. They can be seen in Table 14. The Multi-Task SSL-framework deviates from the other frameworks as it consists of a ResNET structure, pre-trained on the UK Biobank [18] dataset. Furthermore, we implemented a multi-modal approach called *reconstruction* pre-training. Here, the supervisory signal is the ECG signal. A convolutional autoencoder is trained to reconstruct the ECG signal from the BCG signal on the in-house dataset. Given that the dataset for reconstruction pre-training is distinct and significantly smaller, careful consideration is required when comparing these methods.
2. **Augmentations**: We evaluated different augmentations, inspired from Qian et al. [77]. These augmentations were both in the time- and frequency domain. The SSL-model applies two augmentations separately, enforcing the closeness of the encoded versions in the latent space. For testing, we trained each possible pair of augmentations with the *SimSiam*-framework, resulting in 324 pre-trained models. We then evaluated each model by its downstream performance on the Apple Watch dataset. The augmentations are:
   - Time Noise: Adds random noise with mean 0 and standard deviation 0.8.
   - Scale: Amplifies channels with a random factor (mean 2, standard deviation 1.1).
   - Shuffle: Permute the channels randomly.
   - Negate: Inverts the signal values.
   - Permute: Splits signals into up to 5 segments, shuffles, and reassembles them.
   - Resample: Linearly up-samples then down-samples the signal to original dimensions.
   - Rotation: Rotates sensor readings in 3D space by a random angle ($-\pi$ to π).
   - T_flip: Reverses the signal over the time dimension.
   - T_warp: Temporally stretches and warps signals using a cubic spline.
   - Perm_jit: Combines permutation with noise addition.
   - Jit_scal: Applies both jittering and scaling.
   - HFC: Keeps only the high-frequency components after splitting.
   - LFC: Keeps only the low-frequency components after splitting.
   - P_shift: Shifts phase values randomly within $-\pi$ and π.
   - AP_p: Perturbs a segment's amplitude and phase (Gaussian noise for amplitude with mean 0 and standard deviation 0.8, uniform distribution for phase).
   - AP_f: Applies the same amplitude and phase perturbations as AP_p to the entire frequency sequence.
   - *BioInsights*: Detrends the signal, then it runs through a bandpass filter with cut-off frequencies at 4 Hz and 11 Hz, followed by taking the L2-norm and applying another bandpass filter at 0.5 Hz and 2 Hz. Stacks the filtered single-channel signal 3 times to fit with other signals.

3. **Amount of Unlabeled Data for Pre-training**: By varying the amount of data, we can estimate how much data is needed to achieve a well-trained feature extractor. To maintain variety, we randomly sampled a fixed subset of recordings from each subject in the Capture24 dataset. The evaluated amounts of data were 1 %, 10 %, 20 %, 50 %, and 100 %. In addition, we evaluated, how much the performance changes by including all activities instead of just sleeping activities.

4. **Amount of Labeled Data for Fine-tuning**: We varied the amount of labeled data for fine-tuning from 0.1 % to 100 % by sampling a share of samples for each subject in the training set. We then compared the SSL model (NNCLR and SimCLR) against the supervised learning model and their performance on the test set under different amounts of labeled data.

5. **Fine-tune Strategy**: After pre-training the model with one of the different SSL-approaches, we fine-tuned them on the target task. Most works freeze the baseline network and only train the final dense layer on the supervised task. However, as shown by Yuan et al. [109], it can be beneficial to fine-tune the feature extractor's layers along with the last layer. Hence, this work compares the performance of fine-tuning the last layer with fine-tuning the last layer and additional feature extractor layers and evaluates their respective learning rates.

6. **Extracted Features**: To investigate whether the learned features separate the data by different measures in a latent space, we visualized the tSNE plots of extracted layers. tSNE (t-distributed Stochastic Neighbor Embedding) is a statistical method for visualizing high-dimensional data by reducing it to lower-dimensional spaces, typically two or three dimensions, making it easier to plot and interpret visually. It works by converting similarities between data points to joint probabilities and then minimizing the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. This process preserves the local structure of the data, making t-SNE particularly good at creating clusters of similar data points in the reduced space. However, t-SNE can be sensitive to its hyperparameters, like perplexity and learning rate, which can significantly affect the resulting visualization. It is widely used in machine learning for exploratory data analysis, especially to understand the structure of complex datasets like those in genomics or image processing.

Uncertainty Estimation

In order to get a well-calibrated uncertainty estimate, we evaluated different uncertainty models. We implemented all models on top of the baseline supervised model. The implemented models are:

- **Monte Carlo Dropout**: Applying the same dropout rate that was used for training, we collected 100 samples from the model at inference time. The collected samples are then binned into 64 bins to produce a discrete probability distribution, which can be evaluated further.
- **Bayesian Neural Network (BNN)**: We implemented the CorNET model as BNN as in Blundell et al. [6]. Each weight is modeled using a variational Gaussian, with mean μ and standard deviation σ. The means are then initialized with weights from the deterministic trained CorNET, enabling faster convergence. For training, the KL-divergence is minimized together with the MAE-loss. We set the weight prior to $\mathcal{N}(0,1)$, and initalize the posterior to $\mu = 0$ and $\rho = -3$. For inference, the model is sampled 100 times, similar to Monte Carlo dropout, and binned into 64 discrete probability bins. As a preliminary experiment, we only made the first and last layers of the model Bayesian as done by Streli et al. [96]. This, however, resulted in worse performance and calibration.
- **Classification**: A simple classification model is used to classify the samples into 64 discrete bins. The output of the model after the softmax layer is interpreted as probability. For training, each label is modeled as a Gaussian distribution over the 64 bins, with a standard deviation of $\sigma = 3$. We trained the network using the cross-entropy loss.
- **Maxmimum Likelihood Regression**: Inspired by Nair et al. [70], the aleatoric distribution is modeled as a Laplace distribution with parameters μ and s. After the RNN layer, two fully connected heads predict the parameters μ and s for each sample. For training, the negative log-likelihood of the data $D = \{x_i, y_i\}_{i=1}^{N}$ given the parameters θ that parameterize the network μ_θ and s_θ is minimized.

$$
\begin{aligned}
L_{NLL}(\theta) &= \sum_{i=1}^{N} -\log p(y_i|\mu_\theta(x_i), s_\theta(x_i)) \\
&= \sum_{i=1}^{N} \log(2s_\theta(x_i)) + \frac{y_i - \mu_\theta(x_i)}{s(x_i)} \\
&= \sum_{i=1}^{N} \log(2) + b_\theta(x_i) + (y_i - \mu_\theta(x_i))e^{-b_\theta(x_i)} \\
&\text{with } b_\theta(x) = \log s_\theta(x)
\end{aligned}
\tag{18}
$$

  The resulting distribution is then used to compute discrete probabilities over 64 bins, as in the previous models, to ensure comparability and downstream compatibility for post-processing.
- **Deep Ensembles**: We trained the baseline model with 10 different weight initializations to create a deep ensemble of 10 models. Each model is sampled, and the predictions are binned into 64 bins, creating a discrete probability distribution as in Monte Carlo dropout.

The classification and the maximum likelihood regression model both capture aleatoric uncertainty, while the BNN, deep ensemble, and Monte Carlo dropout model capture epistemic uncertainty. Some models work in a regression setting, and others in a classification setting. Since they have different ways of capturing the predictive distributions, all uncertainty estimates were converted into probabilities over discrete bins between 30 bpm and 120 bpm. The amount of bins was chosen

to be 64, as done in *BeliefPPG* [4]. For maximum likelihood regression, the probabilities were computed using the cumulative Laplace distribution. All sampling-based models were sampled into bins, and the classification directly outputs a discrete probability distribution.

Postprocessing

We evaluated three different postprocessing techniques. First, we implemented Kalman smoothing as done by Zhao et al. [114]. It is based on a Kalman filter, which helps to improve the accuracy of heart rate estimates by considering both the uncertainty of the current state and the noise from the measurements. This method refines the estimates by combining the dynamic model of heart rate changes and the characteristics of measurement noise to produce more consistent and precise predictions. The Kalman filter model typically has a few parameters, mainly the process and measurement noise covariances, which are critical for its performance.

Second, we implemented the belief propagation algorithm, similar to Bieri et al. [4]. The underlying structure is a HMM, where the input sample is the observation and the heart rate is the current state. This method updates the estimates sequentially, improving the understanding of the state of the system using a Bayesian approach. The neural network maps the observation to the hidden state, and the transition function is modeled as a discrete conditional probability distribution. It uses a learned log-Laplace function to estimate the probability of the heart rate moving to another state. The process uses forward sum-product message passing (belief propagation), which is effective in updating the state probabilities over time.

Alternatively, inference is executed using the max-product message passing through the Viterbi algorithm. This method is useful for identifying the most probable sequence of states based on observed events. The Viterbi algorithm focuses on the most likely path, which can be beneficial when a clear, specific prediction is needed. Note that both Viterbi and belief propagation do not model the noise explicitly, but work on the underlying probability estimates to find the most likely sequence, they have no parameters to be set. Kalman smoothing models the noise explicitly and has multiple parameters to adjust. Moreover, only the belief propagation operates online, whereas both Viterbi and Kalman smoothing operate offline.

Training

We implemented the models in PyTorch. Training is performed on an NVIDIA Geforce GTX 3090 for 60 epochs unless otherwise stated. We chose the best weights, based on their correlation performance on the validation set. Training a single supervised baseline model takes between 3 and 7 minutes, whereas pre-training a model using SSL takes between 1 h and 3 h. To increase convergence, the heart rate values are normalized, between 0 and 1, where 30 bpm corresponds to 0 and 120 bpm corresponds to 1. We chose these values after analyzing the heart rate values in Figure 3.

Data Quality Metric

In previous work such as by Zschocke et al. [116], a large amount of data was filtered out to improve the average performance. By doing so, one does not directly improve the overall performance on the whole dataset. However, a good metric helps identify those samples where the model fails. For this thesis, we compared five metrics on their ability to identify faulty data. The metrics are:

- **Angle Changes** [101]: For every second, the angle α_k for each axis a_k against the plane constructed by the other two axes a_j and a_l is computed:

$$\alpha_k = \tan^{-1}\left(\frac{a_k}{\sqrt{a_j^2 + a_l^2}}\right) \cdot \frac{180}{\pi}$$

  From these angles, the mean absolute difference over a window is computed. The metric is computed over the maximum of all three axes.

$$m_{angle} = max_k(\frac{1}{N}\sum|\Delta\alpha_k||)$$

- **Maximum Amplitude**: The maximum magnitude over a window length.
- **STD**: The standard deviation of the magnitude over a window length.
- **Mean Value**: The mean magnitude value over a window length.
- **MAD** [116]: The mean magnitude deviation from the windows mean magnitude.

$$|a_i| = \sqrt{a_{x,i}^2 + a_{y,i}^2 + a_{z,i}^2}$$

$$m_{MAD} = \frac{1}{N}\sum_i^N ||a_i| - \overline{|a|}|$$

We conducted two experiments to evaluate which metric is a better measurement to identify faulty data and improve performance. First, a signal processing approach and a deep learning approach predict labels for all samples in the test partition of the Apple Watch and in-house dataset. Then, the absolute error is calculated for every prediction. The five metrics are then computed on the acceleration values of each sample and used to exclude varying amounts of samples from the test set by setting a threshold for values to be smaller.

Followup Experiments

To get a better understanding of how the deep learning model works, we conducted a number of follow-up experiments, mainly aiming at qualitatively analyzing the model and its performance. These experiments include:

- **Frequency Analysis**: Ideally, the trained model acts somewhat similarly to the best-performing signal processing approaches. These apply simple filters in the frequency domain and extract the heart rate from the resulting signal in the time domain. To analyze whether the trained deep learning model uses similar frequency ranges, we implemented a technique coined "frequency occlusion". For this, each sample in the test set is run through a band-stop filter, which removes specific frequencies from the signal. The

resulting signal is then applied to the deep learning model, and deviation from the original prediction is recorded. The full method can be found in Algorithm 1.

Algorithm 1 Frequency Occlusion Analysis

```
Input: test_set, model
Output: response_analysis
Initialize response_analysis as empty dictionary
for each sample in test_set do
    original_pred ← model(sample)
    for each freq_band in frequency_bands do
        filtered_sample ← applyBandStopFilter(sample, freq_band)
        filtered_pred ← model(filtered_sample)
        deviation ← original_pred - filtered_pred
        response_analysis[freq_band] ← deviation
    end for
end for
return freq_response
```

- **Channel Permutation**: The input data consists of 3-dimensional acceleration signals (X, Y, and Z) which represent movement in three-dimensional space relative to the device, usually aligned with a local coordinate system that starts at the device itself—in this case, the user's wrist. Some signal processing algorithms specifically choose a single axis to extract the BCG-signal from. This axis is typically chosen to be the X-axis. However, by doing so, the approach is much less robust to changing environmental conditions, such as rotations, and different sensor models. Ideally, any model should be unaffected by a change of axes, for example, by permutation. Hence, as a second experiment, we measured the deviation in performance due to axis permutation. This can also give an idea about which axis the model pays the most attention to.

- **Channel Sensitivity**: To evaluate how sensitive the model is to different axes, we occlude one of the three channels at inference time. This means, the specific channel is set to 0, while the other two channels remain untouched.

- **Ablations**: We investigated three parameters and their effect on the downstream performance of the supervised deep learning baseline:

  - **Window Size**: Window sizes between 5 and 60 seconds are evaluated. At the same time, the step size is kept constant at 8 seconds.
  - **Step Size**: By changing the step size of windows in the training set, effectively the amount of training data and overlap in training data is varied. The evaluated step sizes are 1 s, 2 2, 4 s, 8 s, and 10 s.
  - **Sampling Rate**: The default sampling rate is 100 Hz. The additionally evaluated sampling rates are 20 Hz, 50 Hz, and 200 Hz. The sampling rate was changed for both the test and training sets.

RESULTS

The following section contains results for the experiments mentioned in Section 3. The results are separated into findings about the dataset, signal processing approaches, and deep learning approaches. The deep learning approaches are again split into supervised learning, self-supervised learning, uncertainty estimation, postprocessing, and data quality assessment. Unless otherwise indicated, all results are reported in mean absolute error (MAE). The results for each component are split up between Apple Watch and the in-house dataset, to get a more differentiated understanding of each component's contributions. While the model's optimization was done on the validation set, the results are reported on the test set, which is independent.

Datasets

First, we analyzed the three datasets described in Section 3.1 by looking at the distribution of heart rate values (Figure 3), derivative of heart rate values (Figure 4), and acceleration values (Figure 19).

The heart rate values are mainly distributed between 40 bpm and 100 bpm, with a long tail towards higher heart rates, as illustrated in Figure 3. The Apple watch dataset has slightly higher mean heart rates (65 bpm) than the in-house (59 bpm) and M2Sleep (61 bpm). The distribution of heart rate values for consecutive 10-second windows looks like a Laplace distribution, as shown in Figure 4. With a standard deviation of 2.4 bpm, 2.2 bpm, and 1.6 bpm, the heart rate changes are centered around zero and are concentrated between -5 bpm and 5 bpm.

In most traces of the Apple Watch and In-House datasets, the Z-axis is aligned with the gravitational axis, but in opposite directions (g and -g). On the M2Sleep dataset, the Z-axis and X-axis are both equally aligned with the gravitational axis.

Analyzing the M2Sleep signal, we found that the value resolution is 0.0156 g on the given dataset. On the other two datasets, the BCG-signal ranges within values of ±0.005 g. Being inside a range smaller than the resolution makes it impossible to capture the BCG-signal on the M2Sleep dataset. Based on these observations, subsequent experiments were conducted solely using the In-House and Apple Watch datasets.

Signal Processing Approaches

While implementing the signal processing approaches detailed in Section 3.4, we observed significant differences in the processing times for a single 10-second window. The BioInsights approach processed a window in 0.026 seconds, whereas the Troika approach required up to 19.1 seconds per window. We attribute the longer processing time for Troika to the computationally intensive task of reconstructing single components extracted using the SSA.

Furthermore, the results in terms of MAE differ strongly between the approaches. While the adapted BioInsights approach achieves 4.40 bpm and 5.51 bpm on the Apple Wacth and in-house dataset, the Troika approach shows the highest error, with 33.37 bpm and 39.92 bpm, respectively. Generally, the

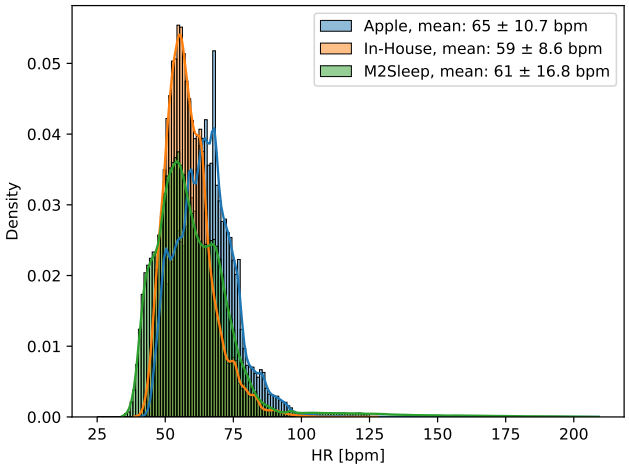

Figure 3: Data - Heart rate distribution for the in-house dataset, Apple watch dataset, and M2Sleep dataset. All datasets show a skewed distribution with means at 59 bpm, 65 bpm, and 61 bpm and long tails towards high heart rates.

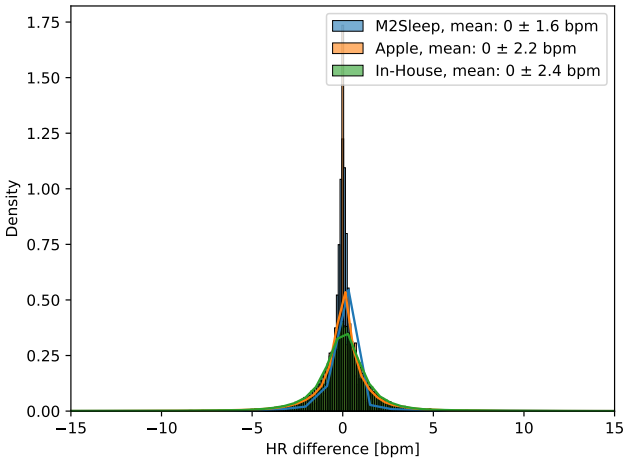

Figure 4: Data - Heart rate differences distribution for the in-house dataset, Apple watch dataset, and M2Sleep dataset. The heart rate changes have a Laplace-like distribution with a mean of 0 bpm.

Approach		Apple Watch MAE [bpm]	In-House MAE [bpm]	Runtime seconds
Median	global	7.40	4.79	
	subject	3.84	3.23	
BioInsights [46]	original	46.95	43.91	0.03
	adapted	**4.40**	**5.41**	0.03
Zschocke [116]	original	15.36	23.27	0.08
	adapted	5.10	7.15	0.08
Steffensen [94]	original	12.94	15.49	0.20
	adapted	7.03	6.67	0.20
Troika [113]	original	33.37	39.82	13.25

Table 4: Signal Processing - Performance of different signal processing approaches, described in Section 3.4, in terms of mean absolute error (MAE) on Apple Watch and in-house dataset.

Hyperparameter	Apple Watch	In-House	Baseline Parameters
Model Architecture	CorNET	CorNET	CorNET
Dropout Rate	0.5	0.3	0.3
Layer Width	64	64	64
Kernel Size	24	8	16
Number of GRU Units	128	128	128
Loss Function	MAE	MAE	MAE
Learning Rate	$5 \cdot 10^{-5}$	$5 \cdot 10^{-3}$	$5 \cdot 10^{-4}$
Batch Size	1024	512	512

Table 5: SSL - Optimal hyperparameters for baseline model in Apple Watch and in-house dataset. The third column contains parameters for downstream experiments.

adapted approaches outperform the original approaches. However, the subject-wise median still achieves the lowest MAE error on both datasets.

Baseline Experiments

After running the experiments described in Section 3.6, we found the parameters in Table 5 to achieve the highest MAE on the test set for each dataset. The third column, called *Baseline Parameters* contains the parameters chosen for all downstream experiments. Further details and each experiment's results can be found in Section 10.1 of the appendix.

Self-Supervised Learning

In this section, we evaluate different self-supervised learning (SSL) approaches, mostly contrastive learning approaches, their augmentations, and the effect of the dataset size and characteristics on the downstream task. We compared all frameworks to the supervised baseline and evaluated them on downstream tasks on the in-house and Apple Watch datasets. The pre-training was done with the Capture24 datasets, and the results are reported as subject-wise 5-fold cross-validation

Dataset	Apple Watch	In-House
BYOL	4.42 ± 0.08	3.68 ± 0.29
Multi-Task	6.88 ± 0.26	3.97 ± 0.72
NNCLR	4.73 ± 0.33	**3.47 ± 0.15**
Reconstruction	4.51 ± 0.25	5.14 ± 0.95
SimCLR	**3.99 ± 0.42**	3.51 ± 0.11
SimSiam	4.08 ± 0.29	3.68 ± 0.29
TS-TCC	8.09 ± 0.85	9.82 ± 5.71

Table 6: SSL - Test MAE [bpm] for different SSL frameworks on the Apple Watch and in-house dataset.

dataset	Apple Watch	In-House
Supervised	4.76 ± 0.33	**3.26 ± 0.24**
1 % sleeping	3.92 ± 0.22	3.33 ± 0.14
10 % sleeping	4.06 ± 0.33	3.38 ± 0.10
20 % sleeping	**3.84 ± 0.31**	3.41 ± 0.10
50 % sleeping	4.99 ± 0.46	3.59 ± 0.15
100 % sleeping	4.68 ± 0.35	3.51 ± 0.12
100 % all activities	4.91 ± 0.38	3.56 ± 0.14

Table 7: SSL - Test MAE [bpm] for different numbers of samples for pre-training on the in-house and Apple Watch dataset.

# Layers	Apple Watch	In-House
1	**4.12 ± 0.36**	**3.62 ± 0.26**
2	4.42 ± 0.43	4.48 ± 0.94
3	4.23 ± 0.41	4.67 ± 1.19

Table 8: SSL - Test MAE [bpm] for different numbers of layers in the last fully connected component on the In-House and Apple Watch dataset.

MAE and its standard deviation on the test splits of both the Apple Watch and in-house dataset. We report the parameters for the final SSL model in Table 14 and Table 15.

Frameworks

In Table 6, we present a comparison of various SSL frameworks based on their MAE scores across two datasets. The reconstruction pre-training framework yields the lowest MAE for the Apple Watch dataset, while the best performance on the in-house dataset was achieved by a model without self-supervised pre-training. It's important to note that all pre-training frameworks utilize the Capture24 dataset, except for the reconstruction framework, which employs the in-house dataset, and the Multi-Task pre-training, which employs the UK Biobank dataset.

We conducted these experiments with the best augmentations from Section 4.4.2 and optimal hyperparameters from Table 14. For further SSL experiment, we use the NNCLR framework since it performed best among all SSL approaches on the in-house dataset.

Augmentations

Figure 5 shows the MAE on the Apple Watch test set for different combinations of augmentations. The lowest MAE-value (3.88 bpm) is achieved with a combination of time warp and *Bioinsights* augmentations. Similarly, time warp and resample achieve good MAE value (3.97 bpm). Third in the ranking is a combination of Low-Pass Filtering and *Bioinsights* (4.01 bpm). Generally, an augmentation that performs well in combination with others is Time Warp, *Bioinsights*, Resample, Low-Pass Filtering, Rotation, Scale, and Permute. Augmentations that achieve low performance in combination with others are Time Flip, Shuffle, Low-Pass Filtering, Noise, and Scale.

Similar results were achieved in the Apple Watch validation set (see Figure 31). Here, however, the best-performing augmentations are a combination of Permute and High-Pass Filtering.

Amount of Unlabeled Data for Pre-training

In Table 7, we evaluate the optimal amount of data for SSL pre-training with the NNCLR framework by taking subsets of the Capture24 dataset from each subject and looking at the downstream task performance. The results are different for both datasets. On the Apple Watch dataset, including 20 % of all pre-training data results in optimal performance. For the In-House dataset, however, performance increases with fewer samples in the pre-training dataset, with the best performance being the supervised approach. Generally, performance is better when only sleeping activity is included, as opposed to including all activities.

Amount of Labeled Data for Fine-tuning

In Figure 6, we investigate the amount of data for fine-tuning and its effect on the model's performance. For sample sizes smaller than 1 %, the SSL approaches (NNCLR & SimCLR) outperform the supervised approach on both datasets. For larger sample sizes, all frameworks converge and perform similarly on their respective datasets.

Fintune-Strategy

In Figure 7, we investigated different learning rates for fine-tuning. While fine-tuning the last dense layer with the default learning rate ($5 \cdot 10^{-4}$), the convolutional and RNN layers are simultaneously fine-tuned with different learning rates. This determines, how much the supervised task loss changes the lower layers at downstream tasks. The best performance for the Apple Watch datasets is achieved when setting the learning rate to $1 \cdot 10^{-5}$, one order of magnitude smaller than the last layer's learning rate. Choosing high learning rates in the order of the general learning rate ($1 \cdot 10^{-4}$ and $5 \cdot 10^{-5}$) for the lower layers is better for the in-house dataset.

Number of Last Dense Layers

We added extra dense layers for downstream tasks, with a hidden size of 256 and 128. These changes did not prove to be effective. Table 8 shows that the lowest MAE was achieved when using a single layer for downstream tasks.

Learned Features

In Figure 32, the tSNE visualizations for extracted features from the last RNN layer can be seen for different pre-training and supervised training stages. For the NNCLR-pre trained

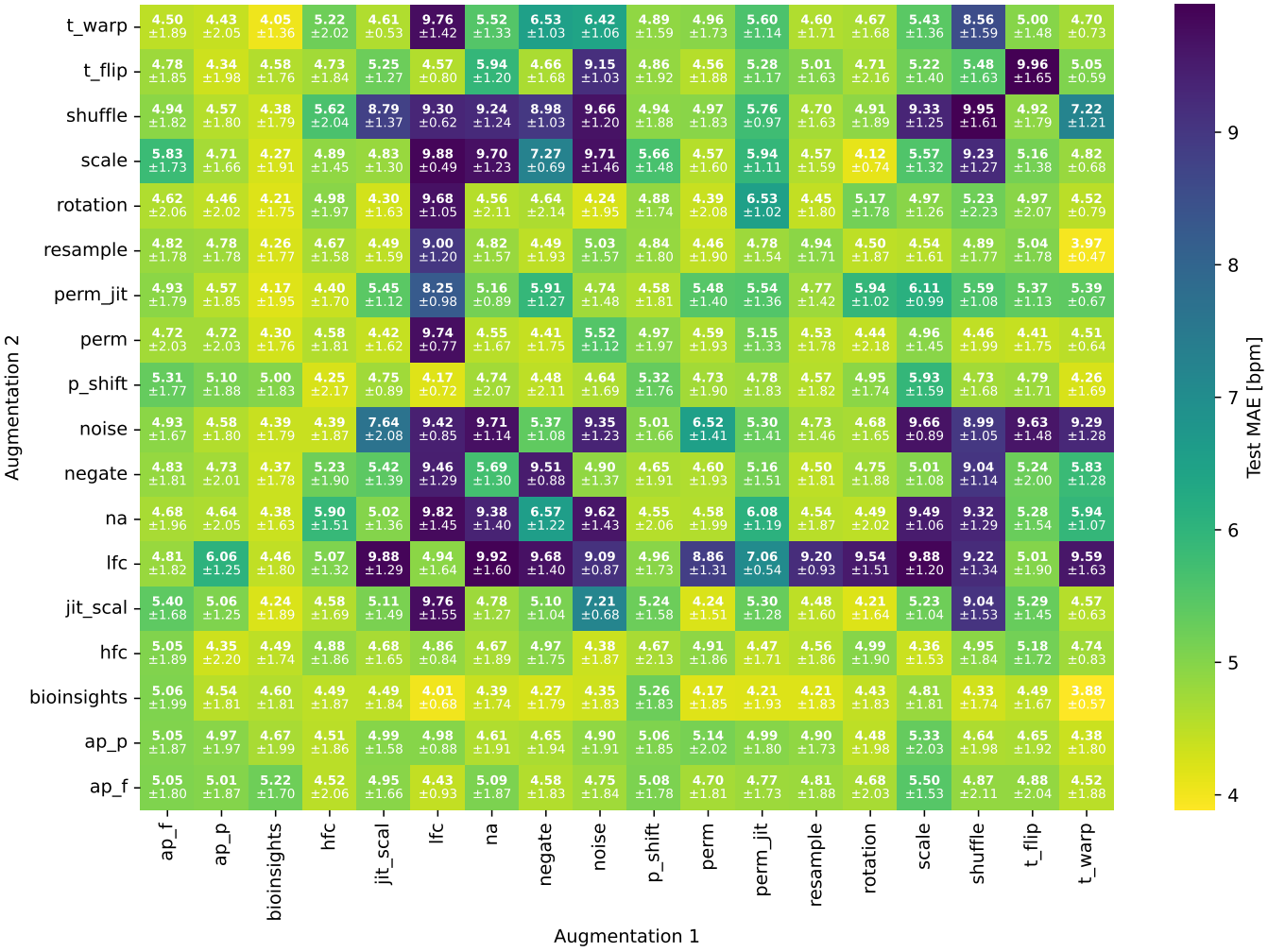

Figure 5: SSL - Comparison of downstream mean absolute error (MAE) in heart rate estimation for different combinations of augmentations. All models were pre-trained with the *Simsiam* framework and fine-tuned with 5-fold cross-validation on the Apple Watch Test set. The x and y axes contain the names of the augmentations (see Section 3.7). *na* stands for no augmentation.

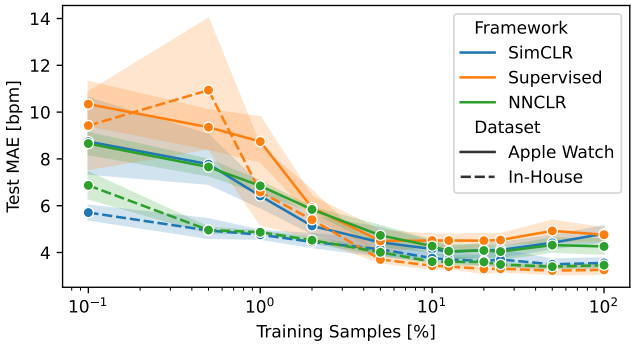

Figure 6: SSL - Amount of labeled data for fine-tuning, for in-house and Apple Watch dataset, and supervised and self-supervised (NNCLR) approach. The x-scale represents the share of data included for fine-tuning in % and is logarithmic. For the Apple Watch dataset, 100 % corresponds to 69k samples and for the in-house dataset it corresponds to 40k samples.

Uncertainty Model	Apple Watch	In-House
Supervised Baseline	4.76 ± 0.33	3.26 ± 0.24
BNN	4.89 ± 0.27	3.29 ± 0.17
Classification	4.92 ± 0.54	3.84 ± 0.65
Deep Ensemble	4.73 ± 0.39	**3.11 ± 0.16**
ML Regression	4.53 ± 0.52	3.25 ± 0.18
MC Dropout	**3.14 ± 0.18**	3.61 ± 0.21

Table 9: Uncertainty - Test MAE [bpm] for different Uncertainty Models on the In-House and Apple Watch dataset.

SSL-model, separation of samples with high heart rate values can be observed, especially in the Apple Watch dataset.

Uncertainty

Some of the uncertainty models sample from a variational distribution, others output the uncertainty explicitly, and others output probabilities over discrete heart rate values. To ensure consistency and comparability, all probabilistic outputs were converted into a discrete probability distribution over 64 bins of possible heart rate values between 30 bpm and 120 bpm. To further evaluate calibration and performance, they are evaluated both numerically, using the MAE and ECE metric, and qualitatively, plotting the performance when removing prediction with the highest uncertainty and plotting the calibration diagram.

As seen in Table 9, the deep ensemble model achieves the lowest MAE on the in-house dataset, whereas Monte Carlo dropout achieves the lowest MAE on the Apple Watch dataset. Further investigation showed that Monte Carlo dropout mainly performs well on a single noisy subject in the Apple Watch dataset (see Table 16). The maximum likelihood regression model performs consistently higher on both datasets than the majority of the other models in terms of mean absolute error. All other methods report similar performance to the supervised baseline.

Uncertainty Model	Apple Watch	In-House
BNN	0.196 ± 0.025	0.137 ± 0.026
Classification	**0.056 ± 0.008**	0.057 ± 0.034
Deep Ensemble	0.440 ± 0.024	0.423 ± 0.027
ML Regression	0.077 ± 0.026	**0.033 ± 0.013**
MC Dropout	0.374 ± 0.006	0.383 ± 0.024

Table 10: Uncertainty - Test expected calibration error (ECE) for different Uncertainty Models on the In-House and Apple Watch dataset.

In terms of calibration, the classification model and the maximum likelihood regression performe best, as seen in Table 10. They report the lowest expected calibration error (ECE) with 0.054 on the Apple Watch dataset for classification and 0.033 on the in-house dataset for maximum likelihood regression. Conversely, the ensemble model reaches the highest ECE-score, being the worst calibrated model. These findings are confirmed when looking at the reliability diagrams in Figure 8 and Figure 33. These diagrams were produced by reducing the bins to 16 and comparing every single probability with the accuracy of the samples in that specific confidence range. The maximum likelihood regression model shows good calibration but tends to be under-confident for high confidence levels. Similarly, the classification model has good calibration for calibrations between 0 and 0.6. It does not predict higher confidences and can, therefore, not be evaluated on these.

Looking at the performance when excluding uncertain predictions in Figure 9, it shows clearly that for both datasets the maximum likelihood regression sorts the predictions by their uncertainty best. Only the classification approach performs similarly. Both approaches model the aleatoric uncertainty, whereas BNN, deep ensemble, and Monte Carlo dropout model epistemic uncertainty.

Postprocessing

For postprocessing, we compared Kalman smoothing with hidden Markov models that use sum-product message passing (Viterbi) and max-product message passing (belief propagation). Both the Viterbi and belief propagation approaches take a distribution over states as input, so we fed them with the probability estimates from the uncertainty models. To find the best combination of uncertainty model and postprocessing approach, we evaluated all combinations on both datasets, as seen in Figure 10. The results vary between the two datasets in terms of best performance. However, when examining the average performance increase by different postprocessing models, the belief propagation model improves the MAE most significantly, by an average of 0.41 bpm, followed by the Kalman smoothing model, which improves it by 0.37 bpm on average. The Viterbi algorithm did not perform well with the Monte Carlo dropout model and deep ensemble model on the Apple Watch dataset. However, it achieves the best average performance on the in-house dataset. Generally, uncertainty models with good calibration show larger performance improvements with probabilistic postprocessing than those with worse calibration. For instance, comparing the MAE of maxi-

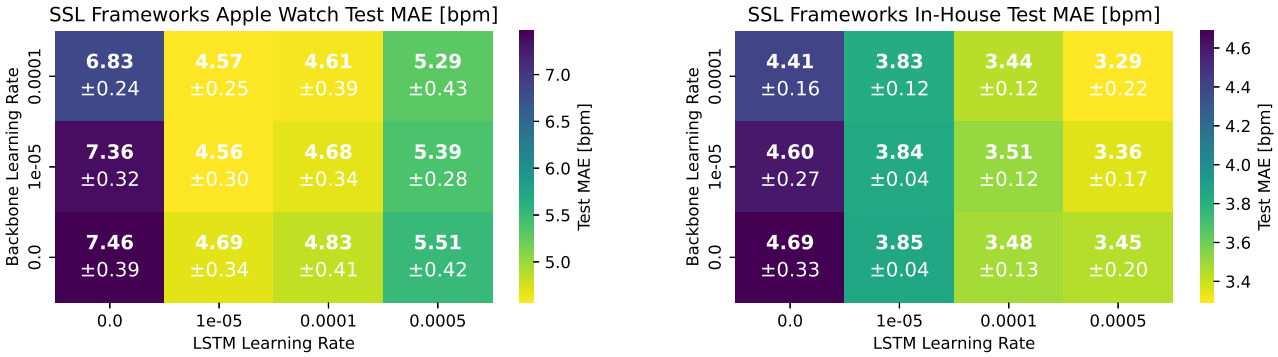

Figure 7: SSL - Comparison of test MAE [bpm] in downstream task for different learning rates in fine-tuning for the in-house (left) and the Apple Watch (right) dataset. The two axes show the learning rates on the backbone convolutional layers and the RNN layers.

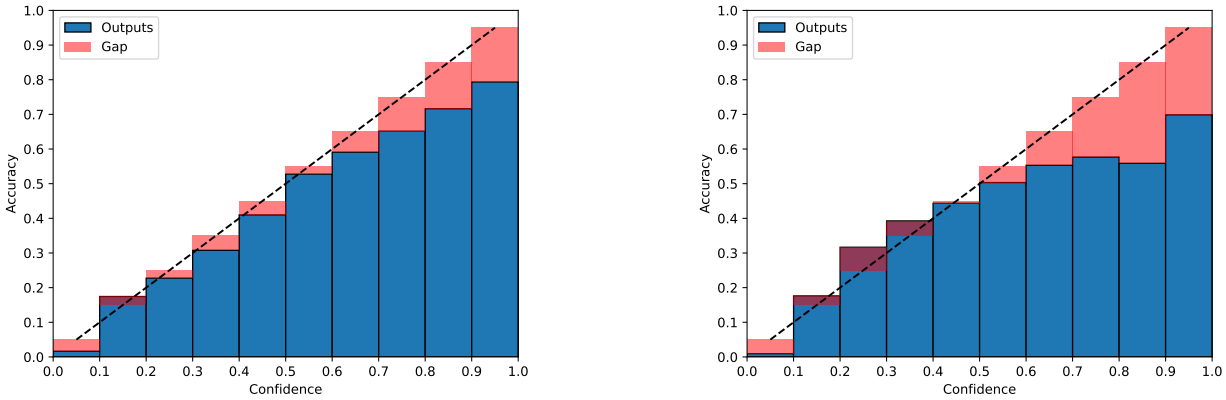

Figure 8: Uncertainty - Reliability Diagram for Apple Watch (left) and In-House (right) dataset. Both depict the calibration for the maximum likelihood regression model. The dotted line represents the perfect calibration goal. The gap represents the distance to perfect calibration for each confidence bin.

mum likelihood regression and deep ensemble in the in-house dataset reveals this trend. Without postprocessing, the deep ensemble model performs better and gains the most from Kalman smoothing. The maximum likelihood regression model, with belief propagation, achieves a performance gain of 0.45 bpm, resulting in the best overall MAE score on the in-house dataset

Overall, a combination of maximum likelihood regression and belief propagation achieves the best MAE score on average. We use it for further downstream experiments and call it the *Uncertainty + Postprocessing* model in the final results.

Data Quality Metrics

We introduced five different metrics to estimate the quality of the accelerometer signals of each sample and their impact on the model's predictive performance. The results appear in Table 11, where we show the MAE when excluding different amounts of data by setting a threshold on the respective metrics. We averaged the results over two datasets and two predictive approaches. The angle change metric performs best, reducing the MAE from 4.3 bpm to 3.8 bpm by just excluding 10 % of data. The angle change metric also has the highest correlation coefficient with the absolute error for the in-house dataset, as seen in Table 11. For the Apple Watch dataset, the standard deviation metric achieves slightly higher correlation coefficients.

	Apple Watch	In-House
Angle Changes	0.21 ± 0.00	**0.15 ± 0.04**
Absolute Max	0.20 ± 0.02	0.11 ± 0.03
STD	**0.22** ± 0.01	0.13 ± 0.03
Mean	0.09 ± 0.02	-0.01 ± 0.02
MAD	0.21 ± 0.01	0.13 ± 0.03

Table 11: Data Metrics - Correlation between different metrics for the accelerometer signal input and the predictive error, averaged over two predictive models, and shown for the Apple Watch and in-house dataset.

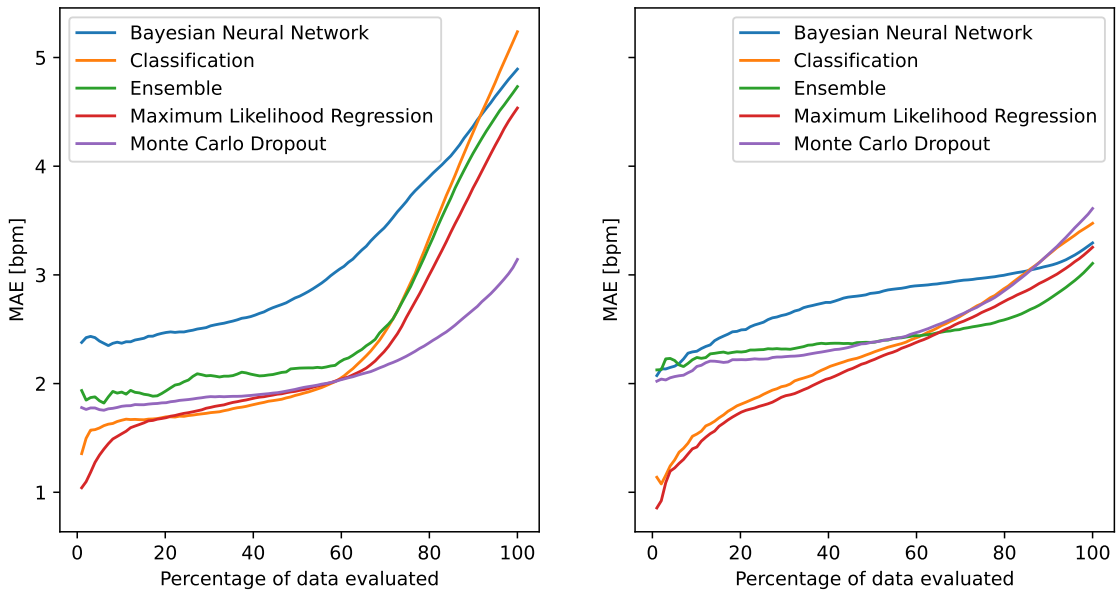

Figure 9: Uncertainty - MAE [bpm] evaluated over different amounts of samples for Apple Watch (left) and in-house (right) dataset. The samples are excluded based on the predicted uncertainty from the corresponding uncertainty models.

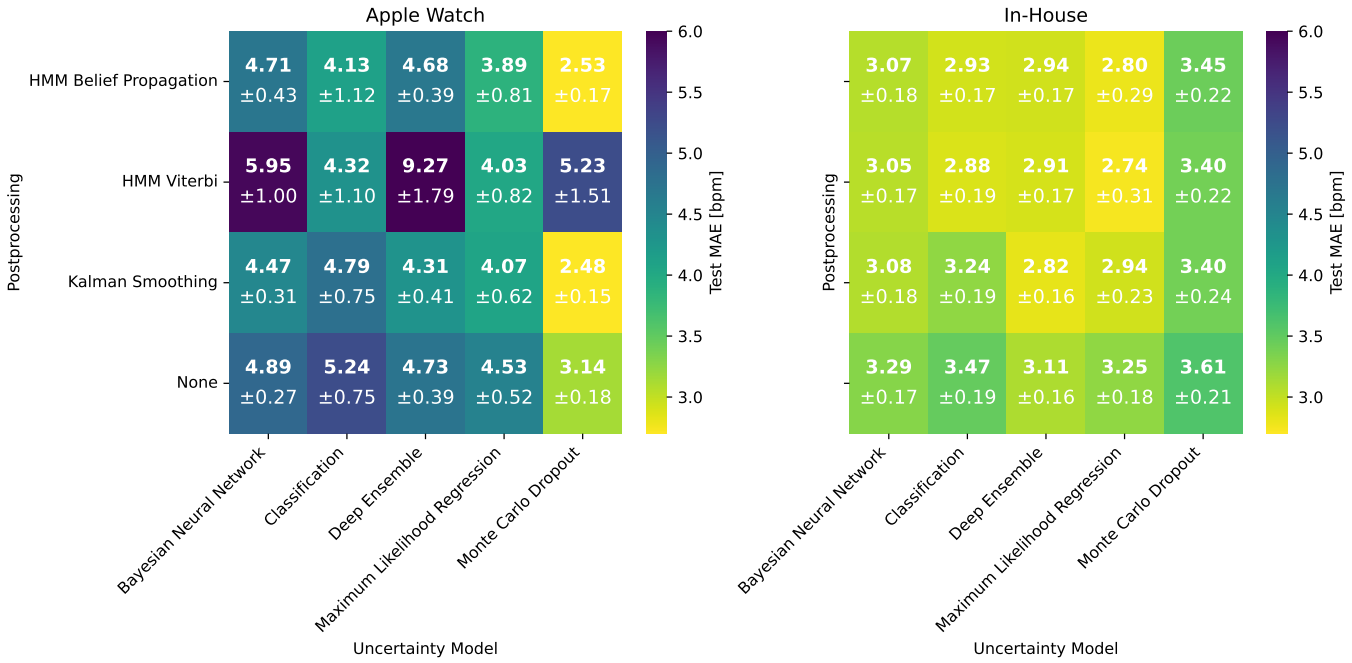

Figure 10: Uncertainty - Test MAE [bpm] for Apple Watch (left) and in-house (right) dataset for different combinations of uncertainty and postprocessing models.

Ablations

Different Ablations were done to further evaluate the format and amount of data that most benefits performance. The full results can be found in Figure 34.

- **Window Size**: The model becomes more accurate with larger window sizes, evaluating sizes from 5 to 60 seconds.
- **Step Size**: By increasing the step size, the amount of overlap in the training data is reduced, thereby the amount of

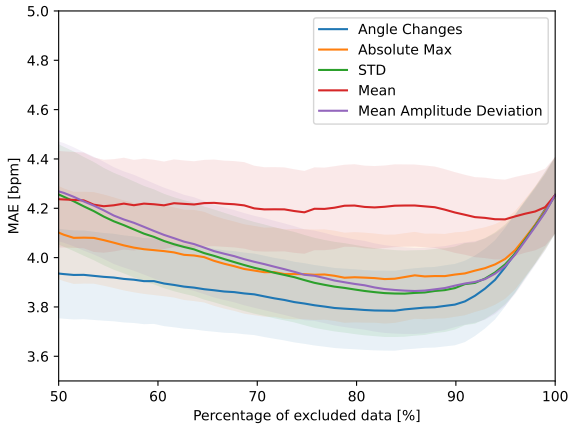

Figure 11: Data Metrics - Test MAE [bpm] for different amounts of excluded data, based on different data quality metrics.

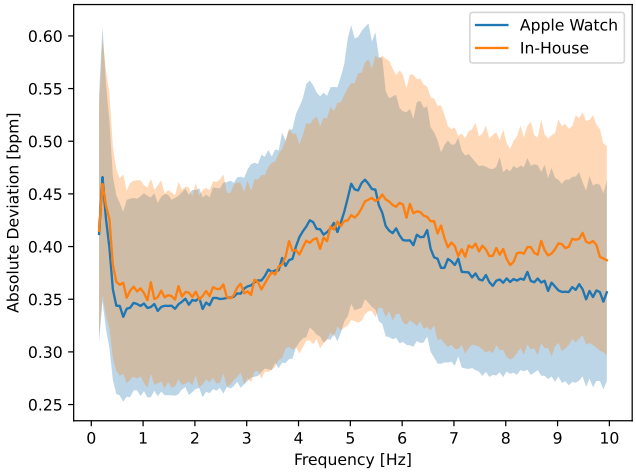

Figure 12: Apple Watch dataset

Figure 13: Frequency Occlusion - Median absolute HR deviation of predictions when applying a bandstop filter for frequencies for Apple Watch and in-house dataset. The plot can be interpreted as a frequency response of the deep learning model. The predictions are done on the baseline model and evaluated for each frequency step.

training data itself. Training with a step size of 10 seconds, which results in no overlap, increases performance in terms of MAE most.

- **Sampling rate**: The sampling rate is varied with values 20 Hz, 50 Hz, 100 Hz, and 200 Hz. For both datasets, 100 Hz is the optimal sampling rate.

Followup experiments

We conducted follow-up experiments to better understand the deep learning model. In detail, these are the follow-up experiments:

1. **Frequency Occlusion**: We convolved a Butterworth bandstop filter of order 4 and a width of 0.25 Hz with the signal for different central frequencies f_c, using a step size of 0.07 Hz. Figure 12 shows the median absolute deviation from the original prediction for different frequencies f_c for the Apple Watch and in-house dataset. There is a peak at approximately 0.21 Hz and a second larger peak at 5 Hz, with two side peaks at 4 Hz and 6 Hz for the Apple Watch dataset. The in-house dataset showed comparable frequency responses but slightly shifted towards higher frequencies. While the prediction deviates strongly with the frequency, the MAE shows a weaker, but similar response.

2. **Channel Occlusion**: We set one of the three X, Y, Z channels to 0, and recorded the deviation from the original prediction without occlusion. The results in Table 12 (upper) show that the prediction deviates on average 1.89 bpm from the original prediction when a single channel is removed. The strongest deviation is when the Y-axis is removed, resulting in a deviation of 2.19 bpm and an increase in MAE to 4.06 bpm on the Apple Watch dataset. On the in-house dataset, the deviations are smaller and more consistent on different axes. As analyzed in Section 4.1, the in-house dataset also has more variance in orientation.

3. **Channel Permutation**: When permuting channels, the prediction deviates almost as strongly as when occluding channels, as seen in Table 12 (lower). Again, the deviations and errors are larger in the Apple Watch dataset than in the in-house dataset. The channel variance is lower, with values ranging from 1.38 bpm to 1.62 bpm overall.

4. **Channel Attention**: The baseline architecture was changed to include a channel attention layer. This layer is inserted after the convolutional layers, and it computes an attention value for each axis of the input signal. The three representations of the input signal's axes are then added, weighting them by their respective attention value. The resulting performance is slightly worse than the baseline approach, decreasing the MAE by 0.12 bpm on the in-house dataset and by 0.14 bpm on the Apple Watch dataset. With the channel attention approach, the sensitivity to channel permutation decreased by an average of 0.17 bpm, the sensitivity to channel occlusion increased 0.8 bpm.

Final Model

We constructed a final model incorporating all previous components and evaluated it on the test sets of both the Apple Watch and in-house datasets. Additionally, we reported the results for each subject in the test set and for the validation set, which are summarized in Table 13. We also computed a median score for each dataset and subject. Specifically, the subject median score calculates the median for each subject and evaluates the MAE by predicting this median value. Conversely, the dataset median predicts all heart rates as the median heart rate from the training set, providing a basis for comparison. In addition to the mean absolute error (MAE), Table 18 reports the result in term of Pearson's correlation coefficient.

Dataset	Apple Watch		In-House	
Channel	MAD	MAE	MAD	MAE
X ↔ Y	1.54 ± 2.5	3.40 ± 4.3	1.37 ± 1.5	3.74 ± 4.0
Y ↔ Z	1.56 ± 2.7	3.39 ± 4.3	1.39 ± 1.5	3.74 ± 3.9
Z ↔ X	1.49 ± 2.5	3.37 ± 4.3	1.62 ± 1.8	3.78 ± 4.0
X = 0	1.49 ± 2.7	3.38 ± 4.4	1.36 ± 1.6	3.79 ± 4.0
Y = 0	2.19 ± 3.9	4.06 ± 5.2	1.30 ± 1.6	3.73 ± 4.0
Z = 0	1.88 ± 3.3	3.74 ± 4.8	1.25 ± 1.5	3.69 ± 4.0

Table 12: Channel Occlusion - MAE [bpm] and MAD from the original prediction on Apple Watch and in-house dataset for channel permutation (upper) and channel occlusion (lower).

Table 13 categorizes the results into baseline and advanced models. The baseline models include the adapted *BioInsights* model and the optimized supervised baseline, as discussed in Section 5. The advanced models feature the Attention model from Section 4.9, the best pre-trained model from Section 4.4, the maximum likelihood regression uncertainty model from Section 4.5, and the maximum likelihood regression uncertainty model combined with belief propagation postprocessing, as detailed in Section 4.6. The last column introduces a combination of the SSL, uncertainty, and postprocessing models. For all models, we report the MAE and its standard deviation for all subjects across both datasets.

The test score for each dataset is the sample-wise mean for all the subjects in the same part of the table. Additionally, we report the validation score. The best model is marked in bold. When comparing the supervised baseline model for different subjects, some subjects achieve much better MAE scores than others. Subject 20 and Subject 8258170 stand out by reporting particularly high errors. These two subjects heavily impact the test score for their respective dataset, which is computed as the mean over all samples. Hence, one should also consider the performance on the single subjects.

On average, the best-performing model is the uncertainty + postprocessing model on the Apple Watch dataset, and the SSL + uncertainty + postprocessing model on the in-house dataset. In total, the SSL + uncertainty + postprocessing model gives the best overall score when combining both datasets. It archives an MAE of 2.84 and 3.08 and a correlation of 0.67 and 0.82 on the in-house and Apple Watch datasets. Furthermore, 75 % of all predictions on the Apple Watch dataset and 79 % of all predictions on the in-house dataset lie within 5 bpm absolute error.

We further analyzed the best SSL + uncertainty + postprocessing model in Figure 14, Figure 35, and Figure 36. Figure 14 shows a Bland-Altmann plot of the predicted heart rate values. For the Apple Watch dataset and the in-house dataset, 95 % and 96 % of all samples lie inside the green lines, presenting a deviation of $1.96 \cdot std$ from the mean differences. For larger heart rates, the model has a bias to predict lower heart rates, and for lower heart rate values, it shows a bias to predict higher heart rates.

Figure 15 shows the heart rate trajectory for subject 844359 from the Apple Watch test set. The ground truth heart rate is shown in blue and exhibits fluctuating behavior with multiple positive spikes along the signal. The prediction by the SSL + uncertainty + postprocessing model and its predicted uncertainty are shown in orange. The predicted heart rate exhibits a less fluctuating behavior than the ground truth heart rate. The uncertainty estimate, however, shows similar spikes as the ground truth. Generally, the predictions follow the heart rate trend well, with single periods of larger deviations, such as the last 20 minutes of subject 844359. Another example is the first hour of subject 6 in Figure 36 of the appendix. Here, the prediction is constantly around 60 bpm, while the ground truth falls from 85 bpm to 70 bpm. Large gaps in the trajectory are explained by the lack of ground truth due to noisy ECG signals.

DISCUSSION

In this section, we evaluate the results reported in Section 4, comparing them to previous work and exploring the implications for further work. As in the result sections, it is split into these categories: Datasets, Signal Processing, Supervised Learning, Self-Supervised Learning, Uncertainty Estimation, Postprocessing, and Further Results.

Datasets

Out of the three proposed datasets for this work, we identified two to be suitable for BCG heart rate estimation. The value resolution of the third dataset (M2Sleep) was too low, and therefore, it could not be used for BCG HR estimation. The two remaining datasets contain a total of 62 participants and 149,771 samples. Comparable PPG datasets, such as the PPG-DaLiA [79] contain 15 subjects and 65k samples. The IEEE Signal Processing Cup 2015 [113] contains 22 subjects but with a total of 3096 windows. Similar HAR-studies work with multiple datasets such as Capture24 with 151 participants [41] and 1.8 million samples, achieving much more statistical power. Furthermore, the datasets that we evaluated in this thesis cover mostly people below the age of 40 while they are asleep. Hence, the results should be viewed within this limited domain.

Exploring the datasets and their distributions in more detail, succeeding heart rate values are highly connected in time, as seen in Figure 4. Enforcing some constraints on changes in the predicted heart rate proves helpful, as we did so by applying postprocessing. Moreover, the distribution of accelerometer values in Table 19 gives a better understanding of the different settings in which the data was recorded. Given the 3-dimensional characteristic of the dataset, we showed that our learning-based model relies on all three axes of the data, as seen in Table 12. This can be seen contrary to related signal processing work, where single axes have been extracted to estimate the target value [94].

Signal Processing

From the four related signal processing approaches, three came from the BCG-domain and one came from the PPG-domain. We adapted all approaches to this task. The adaptions mostly happened after the filtering step and affected the heart rate

Subject	Comparison Model		Baseline Model		Advanced Deep Learning Model				
	Subject Median	Median	Signal Processing	Deep Learning	Attention	Self-Supervised	Uncertainty	Uncertainty + Postprocessing	Self-Supervised + Uncertainty + Postprocessing
2	3.18	3.67	3.63	2.58 ± 0.30	2.64 ± 0.26	2.81 ± 0.09	2.50 ± 0.19	**2.28 ± 0.15**	2.34 ± 0.03
6	3.54	7.64	4.64	2.96 ± 0.38	2.95 ± 0.15	3.75 ± 0.50	3.11 ± 0.59	**2.75 ± 0.62**	3.37 ± 0.72
15	2.65	2.83	3.74	2.21 ± 0.37	2.33 ± 0.20	2.37 ± 0.05	2.11 ± 0.13	1.93 ± 0.32	**1.87 ± 0.07**
20	3.37	4.46	9.22	5.00 ± 0.33	5.00 ± 0.55	4.42 ± 0.48	4.45 ± 0.35	4.01 ± 0.77	**3.51 ± 0.65**
In-House	3.23	4.79	5.41	3.26 ± 0.24	3.29 ± 0.22	3.41 ± 0.10	3.11 ± 0.16	**2.80 ± 0.29**	2.84 ± 0.12
In-House (Val)	-	-	-	3.57 ± 0.32	3.60 ± 0.44	**3.40 ± 0.15**	3.41 ± 0.29	3.59 ± 0.41	3.50 ± 0.23
46343	5.04	15.76	4.52	3.02 ± 0.14	3.01 ± 0.46	3.17 ± 0.42	2.98 ± 0.19	**2.71 ± 0.65**	2.83 ± 0.43
844359	4.54	4.91	2.92	2.52 ± 0.21	2.53 ± 0.09	2.29 ± 0.10	2.41 ± 0.05	2.08 ± 0.31	**1.68 ± 0.15**
4018081	2.29	3.44	2.39	1.92 ± 0.19	1.94 ± 0.18	2.23 ± 0.19	1.83 ± 0.10	**1.31 ± 0.16**	1.50 ± 0.29
8258170	4.46	9.98	7.93	11.44 ± 1.36	12.23 ± 1.45	7.67 ± 1.18	11.64 ± 1.67	9.80 ± 2.72	**6.89 ± 2.16**
9618981	2.61	4.35	3.20	2.58 ± 0.09	2.45 ± 0.09	2.65 ± 0.11	2.35 ± 0.05	1.51 ± 0.14	**1.43 ± 0.06**
Apple Watch	3.84	7.40	4.40	4.76 ± 0.33	4.93 ± 0.35	3.84 ± 0.31	4.73 ± 0.39	3.89 ± 0.81	**3.08 ± 0.53**
Apple Watch (Val)	-	-	-	3.27 ± 0.77	3.21 ± 0.86	3.51 ± 0.83	**2.98 ± 0.79**	3.10 ± 0.69	3.67 ± 0.97

Table 13: Final results - Comparison of MAE [bpm] of different models on the subjects of the Apple Watch dataset and in-house dataset. The signal processing model is the adapted BioInsights model. The deep learning model is the optimized CorNET model. All advanced models extend the baseline deep learning mode. The attention model adds channel attention. The self-supervised learning model consists of a contrastive NNCLR model, pre-trained on the Capture24 dataset. The uncertainty model uses maximum likelihood regression and the postprocessing model uses belief propagation. Combinations of models use the previously mentioned models. All shown subjects are in the test sets of their respective datasets, except for the summarized subjects from the validation set, marked with (val).

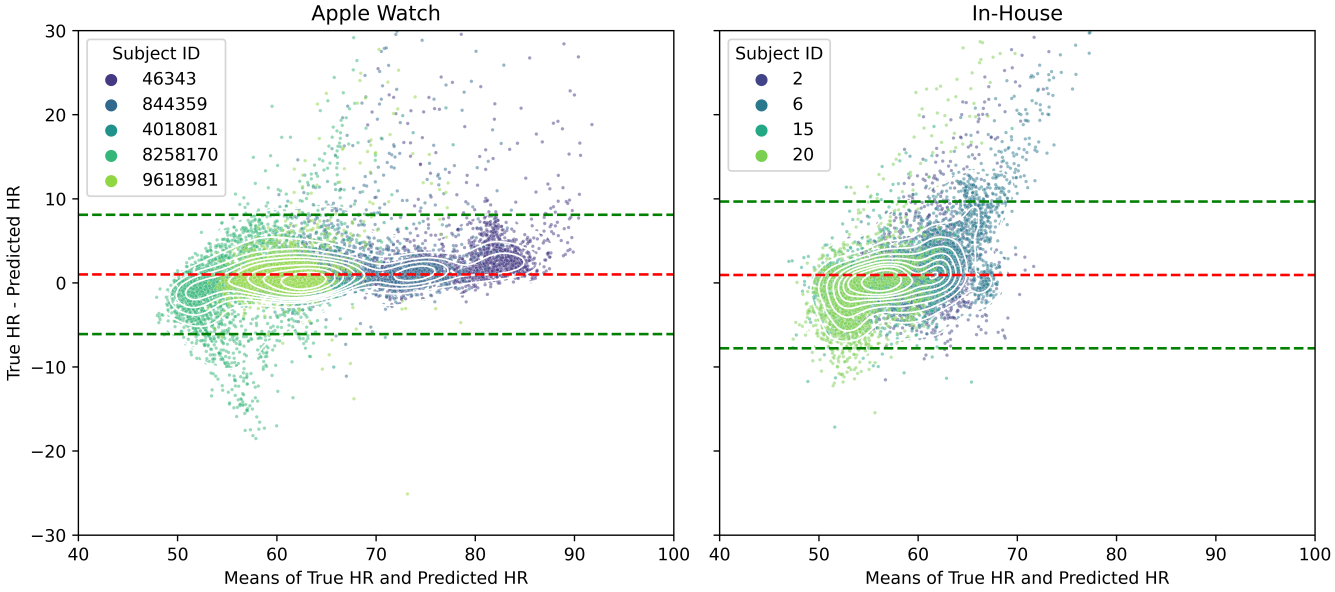

Figure 14: Final Results - Band-Altman plot for the SSL + Uncertainty + Postprocessing model from Table 13 for the Apple Watch (left) and the in-house (right) dataset. The red line represents the mean difference, with the green lines being 1.98 x standard deviation. The colors encode each subject.

estimation from the filtered signal. The BioInsights [46] approach, for example, estimates the heart rate by searching the corresponding peak in the frequency spectrum of the filtered signal. However, this approach achieved very inaccurate results on the given datasets. Moreover, by using a fast fourier transform (FFT)-based spectrum estimation, the heart rate could only be estimated with a value resolution of 6 bpm. We implemented an adapted version that detects the J-peaks in the time domain of the signal using a robust wavelet-based algorithm and that computes the heart rate estimate from the average time between consecutive peaks. This improves the overall performance and achieves the best MAE among all signal processing-based approaches (4.4 bpm).

We also adapted other signal-processing approaches using the wavelet-based heart rate estimation algorithm, which improves performance for the SSA-based approach [94] and the other frequency-domain filter approach [116]. The PPG-domain approach, called *Troika* [113] did not perform well on the

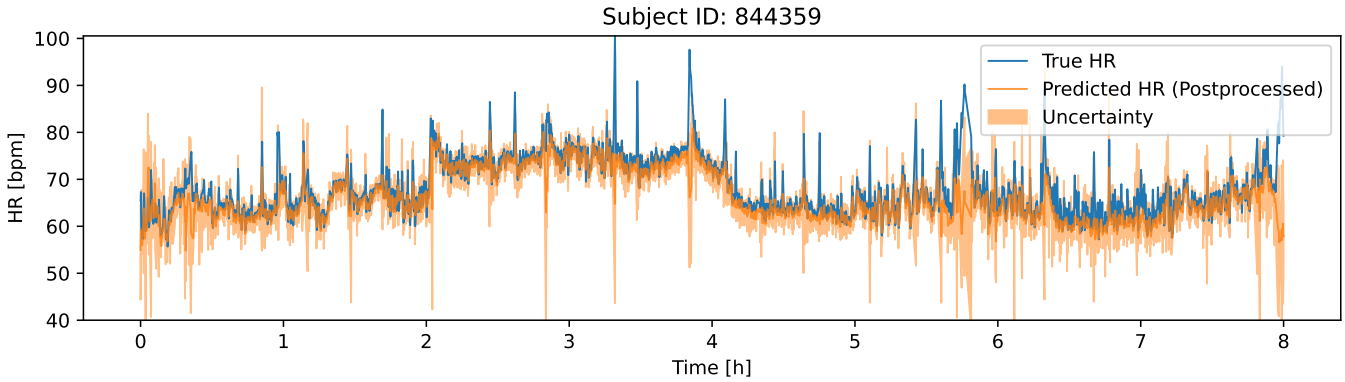

Figure 15: Heart Rate and uncertainty predictions on the Apple Watch dataset for all subjects in the test set, as seen in Table 13. The predictions come from the Uncertainty + Postprocessing model.

given dataset. This could be due to the signal sparsity assumption which might not be fulfilled on accelerometer data. However, since the approach is very complex, and had to be implemented in Python, it could also be due to errors in the implementation or a lack of parameter optimization. The *Troika* approach, however, is also the most computationally expensive approach, computing a SSA-decomposition and reconstruction the signal for every single sample. Except for the approach by Steffensen et al. [94], all signal-processing-based approaches operate on the L2-norm of the 3-dimensional signal and, therefore, lose potentially useful details in the composition of all three axes.

Although relatively simple, the signal processing approaches are computationally expensive and have a large parameter space to optimize. Hence, in the further investigation of this work we evaluated learning-based models, namely Neural Networks. They have shown superior performance in similar tasks, such as PPG heart rate estimation (see Table 3).

Supervised Learning

We adopted a supervised learning approach as our initial baseline, utilizing the Apple Watch and an in-house dataset. We evaluated different architectures and hyperparameters and established a baseline set of hyperparameters (refer to Table 5). In these baseline experiments, the CorNET model, originally designed for PPG-based heart rate estimation, demonstrated the best performance. Further testing revealed that reducing the kernel size and the number of convolutional kernels per layer surpassed the performance of more complex architectures. However, when we trained the *HRCTPNet* model, it failed to converge to a satisfactory solution on the Apple Watch dataset and performed poorly on the in-house dataset. CorNET outperformed other architectures such as simple GRUs or fully convolutional networks. Despite its relatively shallow structure and modest parameter count (208k) compared to, for instance, *HRCTPNet*, it proved to be the most effective solution for a baseline deep learning task in heart rate estimation. This superiority may stem from the fact that deeper and larger networks are more susceptible to overfitting. A shallow architecture with fewer parameters, along with regularization techniques, can mitigate this issue while effectively extracting the necessary features from the data. After testing three different loss functions (MAE, MSE, Huber), the MAE loss function achieved the highest MAE-score, which was expected as the model optimizes this metric. However, also the correlation between true and predicted heart rate values confirmed that the MAE loss function outperformed the others. The MSE loss function is highly sensitive to outliers, whereas the MAE loss function largely ignores them, offering much greater robustness. The optimal dropout rate ranged between 0.3 and 0.5, underscoring the benefit of actively preventing overfitting to enhance the network's performance in the task.

The supervised learning approach already performed reasonably well and achieved with 3.26 bpm a higher MAE on the in-house dataset than the best signal-processing approach (5.41 bpm). On the Apple Watch dataset, however, it performs with 4.76 bpm slightly worse than the signal processing approach (4.40 bpm).

Self-Supervised Learning

For self-supervised learning (SSL), we set a number of best hyperparameters, including augmentations, framework, and fine-tune strategy. The best model performs well on subjects with noisy data and achieves the lowest MAE on the in-house validation set. However, it also lacks a general advantage over the supervised baseline, revealing higher errors for single subjects.

When comparing different SSL-frameworks, contrastive learning frameworks outperform others like generative or multi-task frameworks on this dataset. For contrastive learning, the model usually learns to contrast different augmentations of the same sample. Among the best augmentations was the *BioInsights*-inspired augmentations. It extracts a frequency-filtered signal as in *BioInsights* [46]. Generally, however, there is a large variance in the performance of each combination of augmentations. This indicates that a general statement can not be made as to which augmentations work best. Similarly, Qian et al. [77] concluded that there is no uniquely best augmentation for SSL on HAR. However, it can be deduced that low-frequency components, shuffle, and noise prevent the

model from learning meaningful representations, impairing the downstream performance.

Furthermore, we evaluated what characteristics and amount of unlabeled data are needed for good downstream performance. Again, the two datasets do not share the same result. On the in-house dataset, the downstream performance worsens with SSL-pre-training, while on the Apple Watch dataset, 20 % of the data in Capture24 is optimal for downstream performance. This indicates that, especially for the in-house dataset, the differences between the two domains are too large to benefit from downstream performance in the case of the in-house dataset. Including all activities from Capture24 in the pre-training set further degrades performances, which could be explained by the larger domain shift between the HAR Capture24 dataset and our datasets.

When evaluating the impact of data quantity on fine-tuning, the significant advantage of the SSL model becomes evident, as illustrated in Figure 6. Specifically, when fine-tuning is performed with limited data (approximately 1,000 samples), the SSL pre-trained model consistently surpasses the performance of the supervised learning baseline, which is trained with an equivalent amount of data. However, this performance advantage of the SSL model diminishes as the sample size increases.

The fine-tuning experiments show that it is crucial to also fine-tune the RNN layers inside the backbone model, as seen in Figure 7. However, it is not necessary to fine-tune the convolutional layers. This agrees with the findings by Yuan et al. [109] but stands in contrast with other models like Haresamudram et al. [42], where only the last layers were fine-tuned.

Furthermore, the model does not benefit from additional fully-connected layers. This suggests that a single dense layer is complex enough to transform the extracted features into reasonable predictions. Haresamudram et al. [42] showed similar results for HAR, where the additional encoding layers degrade the performance.

From Table 13, we understand that contrastive SSL mostly improves performance on samples where the supervised baseline performs poorly, whereas on easy samples it usually performs worse than the supervised baseline. A possible interpretation is that the SSL guides the model to a different, more robust solution than the supervised baseline. Hendrycks et al. [43] similarly found that self-supervised learning help the model be more robust to label corruptions and adversarial training.

Uncertainty Estimation

We compared different models for the uncertainty estimation. The classification and the maximum likelihood regression model both capture aleatoric uncertainty, while the BNN, deep ensemble, and Monte Carlo dropout model capture epistemic uncertainty. Each distribution was transformed into a distribution over discrete bins. While this enables the modeling of multimodal distributions, the heart rate discretization introduces quantization errors.

Sampling-based models and regression models both outperform the classification model in terms of MAE. The deep ensemble model especially achieves low error rates, ranking first on the in-house dataset and second on the Apple Watch dataset. This is in agreement with previous works on PPG heart rate estimation, where model ensembles were used to improve performance [12, 79].

In terms of expected calibration error (ECE), the maximum likelihood regression and the classification have the lowest ECE, as seen in Table 10, indicating better calibration than the other models. It is interesting that both these models model the aleatoric uncertainty. For a complete uncertainty estimate, both aleatoric and epistemic uncertainty should be modeled as done by Amini et al. [2]. However, these findings suggest that in our case the aleatoric uncertainty outweighs epistemic uncertainty. Epistemic uncertainty often comes through a lack of training data or out-of-distribution samples. Aleatoric uncertainty can be understood as data uncertainty, for example in the label. Through the small window length, only a few heartbeat R-peaks are available to compute the ground truth heart rate. This entails a large variance in the labels and could explain the larger aleatoric uncertainty.

The reliability diagram confirms the findings from evaluating the expected calibration error (ECE). The maximum likelihood regression shows a good calibration except for high-confidence areas, where it tends to be overconfident. All other models exhibit similar behavior and are overconfident in high-confidence areas but are also calibrated worse in lower-confidence areas, as shown in Figure 9.

For all uncertainty models, the MAE is strictly monotonically decreasing when excluding the predictions with the highest uncertainty. This indicates that for all models the calibration can be improved by simple uncertainty calibration measures such as Platt scaling or isotonic regression [38]. However, the maximum likelihood regression and classification best identify samples with low predictive error.

Postprocessing

We designed the postprocessing to enforce the inherent temporal connectivity between succeeding samples and their respective heart rate labels. We implemented three postprocessing models, namely belief propagation, Viterbi, and Kalman smoothing. All three assume a hidden Markov model (HMM)-like structure with the heart rate estimate as the hidden state and the acceleration signal as the observation. Viterbi and Kalman smoothing operate offline, having a forward and a backward pass, whereas belief propagation operates online, with only a forward pass. Viterbi and belief propagation estimate the probability of the heart rate falling into different states (heart rate bins) while putting no assumption on prior states and taking the probability outputs from the uncertainty model. Kalman smoothing runs on the continuous heart rate estimate, modeling the observation and process noise as Gaussian distributions.

As shown in Figure 10, the Kalman smoothing algorithm performs well with all underlying uncertainty models, averaging around 0.375 bpm improvement in MAE. Since it does not con-

sider the underlying model's uncertainty estimates, but only works on the HR estimates, it simply helps smoothing out the prediction. The improvement agrees with Zhao et al. [114], where Kalman smoothing improves the MAE by 0.58 bpm and 1.47 bpm in two experiments.

The Viterbi and belief propagation algorithms rely on uncertainty estimates and are highly affected by badly calibrated models. This becomes clear when looking at the average improvement of these models for BNNs, deep ensembles, and Monte Carlo dropout models, combined with the Viterbi algorithm. On average, the performance is decreased by 1.17 bpm. However, when combined with a well-calibrated uncertainty model, like the maximum likelihood regression or the classification, the Viterbi algorithm improves the MAE by 0.64 bpm on average. The belief propagation algorithm is less affected by bad calibrations, averaging 0.36 bpm and 0.68 bpm increase in MAE on the two datasets. We explain this by the fact, that the Viterbi algorithm outputs the most likely trajectory of states given the observations, whereas belief propagation outputs soft probabilities over multiple states. The heart rate is then predicted by computing the expectation over the discrete distribution instead of taking the mode of the distribution, which allows for smoother estimates.

Data Quality Metrics

We evaluated different metrics to assess the quality of an accelerometer signal sample to filter out samples that result in bad predictions. The best metric computes the absolute amount of changes in the rotation of its axes. This indicates that situations where the subject rotates or moves their arm, such that the gravitational axis changes, mostly affect the prediction quality. Other metrics, such as the one used by Zschocke et al. [116] (MAD), show similar performances when filtering out less than 10 % of all samples. However, when excluding more than 10% of all samples, the MAE increases strongly for MAD and STD, resulting in higher error rates than with 100% of the data. In the dataset by Zschocke et al. [116], this metric was used to exclude 74.3 % of all samples, which would result in high errors on this dataset.

Ablations

The model's performance benefits from a large step size (10 s), a large window size (60 s), and a medium sampling rate (100 Hz). The effectiveness of the large step size suggests that overlapping training data does not enhance the model's learning. It is crucial to interpret the optimal sampling rate of 100 Hz with caution, as the baseline model was specifically optimized for this rate. Additionally, the in-house dataset inherently uses a sampling rate of 100 Hz. The improved downstream performance due to the large window size can be attributed to two factors: firstly, the predictions become more accurate because the model processes more data before making a decision; secondly, averaging more heartbeats per sample window reduces noise in the data labels.

Followup Experiments

As we showed in the follow-up experiments, the model is sensitive to permutations and occlusions of channels and responds most to frequencies around 0.2 Hz and 5 Hz. Intuitively, the model should pay attention to the frequency areas where the heart rate lies, which is around 1 Hz. As shown by Yao et al. [107], most of the energy of the BCG signal lies in its harmonics, with the 2nd to 5th harmonic explaining 88 % of the variance in the signal. This coincides with the peak around 5 Hz and its side-peaks at 4 Hz and 6 Hz. The peak around 0.21 Hz has two possible explanations. Firstly, the neural network is sensitive to offset changes in the signal. By filtering out low frequencies, the offset is effectively removed. However, in the experiments, the signal was z-normalized after band-stop filtering. A second explanation is that the peak corresponds to the respiratory frequency. This usually lies between 12 bpm and 20 bpm during sleep [23], which corresponds to 0.2 Hz to 0.33 Hz. Hence, the network shows signs of paying attention to the respiratory rate in addition to the BCG signal to estimate the heart rate. In a downstream experiment, where the network was trained on a band-pass filtered signal, removing frequencies below 0.5 Hz resulted in significantly worse performance, which confirms the assumption that low frequencies are essential for the model to perform well.

Final Results

There is no model that performs best on all datasets, as seen in Table 13. However, when combining the previously mentioned and evaluated components, such as self-supervised learning (SSL), uncertainty, and postprocessing, the resulting performance exceeds the performance of the isolated components in most cases. However, the resulting performance in terms of MAE does not exceed the subjects' median in all cases, and still leaves room for improvements.

Furthermore, the Bland-Altmann plot (Figure 14) shows a model's predictive bias towards heart rate values around 60 bpm. This could be expected, due to the low variance in the training data heart rates. The bias is expected to be stronger for subjects with exceptionally high or low heart rates. A possible solution is further data augmentation or a customized loss as used by Spathis et al. [90].

Key Takeaways

Different outcomes have been learned through the course of this thesis. They are summarized here.

1. **Signal Processing**: All evaluated signal processing approaches provided unsatisfactory results for the task of wrist-BCG. Our adaptations using time-domain continuous wavelet transform lead to significant improvements. Ultimately, our adaptation based on the *BioInsights* approach gives a robust baseline but is computationally inefficient and lacks performance in comparison to learning-based methods.

2. **Data Quality Metrics**: Out of all compared data quality metrics, the angle change metric identifies motion artifacts in the input signal best, finding samples with axis rotation during the length of a window. It can be used to filter out samples for which the presented models struggle at heart rate estimation.

3. **Preprocessing**: A large window size of 60 seconds improves the model's performance. This, however, comes with a trade-off in model size and inference speed. A step size of 10 s maximizes performance, indicating that the mode does not benefit from overlapping samples when training with small window sizes.

4. **Supervised Learning** CorNET, a shallow CNN-GRU network outperforms other network architectures. The MAE-loss function, in combination with strong regularization, is robust to noisy training data. Together, these components build a supervised baseline and outperform signal-processing approaches.

5. **Self-Supervised Learning**: Contrastive learning outperforms other SSL-models, with our proposed BCG-inspired bandpass-filter augmentation boosting performance. It does not clearly beat the supervised approach, possibly due to the large domain gap. However, it helps the model to perform more robustly, performing well on specific subjects where the supervised model surprisingly fails. Furthermore, it outperforms supervised learning in situations with few labeled training samples.

6. **Uncertainty**: Maximum likelihood regression, an approach that models aleatoric uncertainty, shows the best calibration among different evaluated uncertainty models. However, deep ensemble achieves the lowest overall error.

7. **Postprocessing**: Belief propagation based on the output of maximum likelihood regression is the best-performing combination of uncertainty model and postprocessing. However, probabilistic postprocessing approaches such as belief propagation and Viterbi fail when used with badly calibrated uncertainty models. Kalman smoothing, however, is the only postprocessing technique that improves the performance reliably and independently of the uncertainty calibration, but has many parameters to optimize.

8. **Final Model**: The combination of SSL, uncertainty, and postprocessing generally outperforms its components and the supervised baseline. The resulting model predicts 75 % and 79 % of all test samples within 5 bpm of their true heart rate value on the Apple Watch and in-house dataset. The mean absolute error is 2.84 bpm and 3.08 bpm, and Pearson's correlation coefficient is 0.85 and 0.67 on the in-house and Apple Watch datasets, respectively.

9. **Frequency Sensitivity**: The deep learning model is frequency selective. It is especially sensitive to frequencies in high-energy areas of the BCG-signal (4 Hz - 6 Hz) and around the frequencies corresponding to the respiratory rate (0.21 Hz).

10. **Channel Sensitivity**: The deep learning model uses all three axes of the acceleration signal to predict the heart rate. It is sensitive to permutations and occlusions of the axes' signals. Hence, it is also sensitive to rotations of the sensor itself.

LIMITATIONS

In this section, we critically examine the constraints of the methodologies and results presented previously, addressing the limitations of individual components before considering the system as a whole.

One primary limitation is the data availability. The employed datasets includes a restricted number of healthy participants, confined to a narrow age range, and data collected from a single device location, namely the wrist. Consequently, the models developed are primarily applicable within these specific conditions. Since the BCG signal is affected by age and cardiac conditions, it is likely that the models do not work outside these populations and would need to be restrained to give meaningful results.

Moreover, the subset of the data used for testing was limited to only nine subjects, which may not adequately represent the wider population. Although the test set subjects were chosen to represent the whole dataset as well as possible, this limitation still reduces the generalizability of the findings.

Additionally, the ground truth labels for the Apple Watch dataset were derived from photoplethysmography (PPG) data, with no detailed information on the data processing methods or the reliability of these measures. This uncertainty could affect the accuracy of the models based on this dataset.

A significant methodological gap noted is the absence of a benchmark dataset or comparable methodologies for this specific task. This absence makes it difficult to evaluate the relative performance of the proposed approach against existing methods.

Furthermore, the analysis lacks a comprehensive qualitative assessment that could provide deeper insights into where and why the model succeeds or fails. Such an evaluation would be crucial for refining the model and enhancing its applicability and reliability. However, this is a general problem in the state-of-the-art of current machine learning models.

FUTURE WORK

In this section, we outline potential directions for further research based on the findings and limitations of this thesis. We investigate gaps in the current thesis and suggest strategies to address those.

While the approach is currently only applied to wrist-worn sensors, it could be applied to more diverse sensor locations. Hernandez et al. [44] previously applied a similar signal-processing-based approach to head-worn devices. The concept could be extended to smartphones that are either carried on the body or placed on the bed next to the sleeping person. They could also include a gyroscope's signal, since it is usually included in IMUs that measure the acceleration. Further analysis could identify different heart conditions or a person's age from the BCG signal. However, there is no corresponding data available at the moment.

Additionally, a comparative analysis of the performance of BCG-derived heart rate and PPG-derived heart rate would give a better understanding of failures and possible use cases of the

current approach. This requires a dataset with ECG, PPG, and BCG recordings while asleep.

The dataset presents further opportunities for analyzing different physiological signals, such as sleep stages and heart rate variability (HRV), given that the in-house dataset has accurate ECG measurements and the Apple Watch dataset has sleep stages, derived from polysomnography. Since both physiological signals interact with BCG-derived heart rate, the existing model's weights could be used in a transfer-learning fashion.

Furthermore, future research could develop a generative model to generate a clean ECG signal from the BCG signal, as done by Sarkar et al. [84] for the PPG domain. First steps of this have been taken in this thesis when constructing a generative model as a self-supervised learning (SSL) pre-training task. However, due to low downstream performance, these were not investigated further. The multimodal character of the data could also be used to pre-train the model, similar to *ColloSSL* [51].

The results have shown that the model is sensitive to axes rotation and permutation. However, with real-world data, these artifacts cannot be avoided. Further steps should be taken to increase the model's robustness to changes in the axes. These steps can lead to a better understanding of the utility of different axes and can result in a smarter way to fuse the three axes, for example, using attention mechanisms. A first step in this direction has already been taken, but further research is necessary.

Further in-depth analysis on larger demographic groups, including people above the age of 40, people with cardiac conditions, and children, could further increase knowledge of the model's generalizability. Private datasets, such as the *Fenland* study [33] or the dataset used by Zschocke et al. [116] could mitigate this issue.

Finally, the developed model could be applied to large, unlabelled datasets, such as the UK Biobank [18], to test whether a reasonable heart rate estimate can be derived for this dataset. Further analysis could include the interaction between sleeping heart rate and health-related outcomes.

CONCLUSION

This thesis was aimed at estimating heart rate during sleep from wrist-worn BCG using deep learning. Based on the quantitative result on this limited dataset, it was shown that a deep-learning-based heart rate estimation is possible, with 75 % (Apple Watch) and 79 % (in-house) of all samples being within 5 bpm of the PPG and ECG-derived heart rate on two evaluated datasets.

We presented an extensive evaluation of techniques for wrist-worn Ballistocardiography-based heart rate estimation while asleep. After comprehensive literature research, we implemented multiple signal processing-based and deep learning-based approaches and compared them on two datasets. The initial signal processing model averages a mean absolute error (MAE) of 4.9 bpm. Subsequently, we implemented a supervised learning model that achieves 4.0 bpm on both datasets. We improved this further by using self-supervised learning (SSL) and uncertainty estimation together with probabilistic postprocessing. The resulting model reliably predicts the heart rate on the given dataset for sleeping subjects, achieving an average MAE of 2.96, and an average Pearson's correlation coefficient of 0.76 on both datasets. Further analysis was conducted, analyzing the sensitivity and qualitative performance of the model and showing that the model acts similarly to frequency-based approaches when estimating the heart rate.

There are three key findings

1. The data must have an accelerometer value resolution sufficiently large to capture BCG-signals in the range ±0.005 g. Filtering out noisy samples with motion artifacts or rotations can improve downstream performance, and using large window sizes helps to smooth out the ground truth heart rate and adds information that increases downstream performance.

2. No deep-learning configuration outperforms all other possible configurations. Its optimal configuration depends on the dataset and varies from subject to subject. However, in general, a CNN-GRU model with contrastive pre-training on unlabelled data, maximum likelihood uncertainty estimation, and belief propagation maximized performance on the given dataset and outperforms signal-processing-base approaches.

3. The extracted model learns to extract features from frequency components that correspond to the BCG signal, indicating that it acts similarly to a complex frequency filter. It is sensitive to channel permutations and has a predictive bias towards median heart rates.

These findings suggest that heart rate can be estimated by a wrist-worn accelerometer under specific conditions like sleeping using real-world data. They raise the question of how well the model generalizes on large-scale datasets, such as the UK Biobank and different body positions and devices. Furthermore, they open the possibility of analyzing accelerometer traces from more datasets by extracting the heart rate.

ACKNOWLEDGMENTS

I am deeply thankful to Professor Dr. Christian Holz and my supervisor Max Möbus for their mentorship and support during my research journey. I also appreciate the valuable feedback from everyone at SIPLAB.

A special acknowledgment goes to my girlfriend, Molly, for her constant support, and to my family for their encouragement and love.

APPENDIX

Deep Learning Baseline Experiments

The deep learning baseline was found by running 5-fold subject-wise cross-validation and evaluated by looking at the MAE on the test set. The results are shown in Figure 20, 21, 22, 23, 24, 26, 28, 29. Each plot shows the mean test MAE over 5 folds and its standard deviation. The results are summarized in Table 5.

Architecture

The CorNET architecture with 208k parameters performs best on datasets, followed by the GRU-architecture with 150k parameters.

Model Parameters

The model parameters were evaluated individually as deviating from an initial set of parameters found by running a Bayesian optimization with a single fold.

Different convolutional kernel sizes are compared in Figure 21. The best-performing kernel size is 24 for the in-house dataset and 8 for the Apple Watch dataset.

Figure 22 compares the number of kernels per convolutional layer in the CorNET architecture. For both datasets, 64 kernels per layer performed best.

Figure 23 evaluates the amount of RNN-unit in the CorNET architecture. In both datasets, 128 RNN units yield the lowest MAE.

Different dropout rates are evaluated in Figure 24. On the in-house datasets, a dropout rate of 0.5 performs best, whereas, on the Apple Watch dataset, a dropout rate of 0.3 performs best on average.

Training Parameters

Figure 26 compares different loss functions. The MAE-loss yields the lowest MAE-value on both datasets. The loss function is also evaluated on the correlation metric in Figure 27, showing similar results.

Large batch sizes outperform smaller ones as shown in Figure 28 on the in-house dataset, with a batch size of 512 results in the lowest MAE, and on the Apple Watch dataset, a batch size of 1024 results in the lowest MAE.

Figure 29 shows the MAE for different learning rates. Generally, a higher learning rate yields a better performance for the in-house dataset, with $5 \cdot 10^{-4}$ being the best. For the Apple Watch dataset, $5 \cdot 10^{-5}$ yields the best performance.

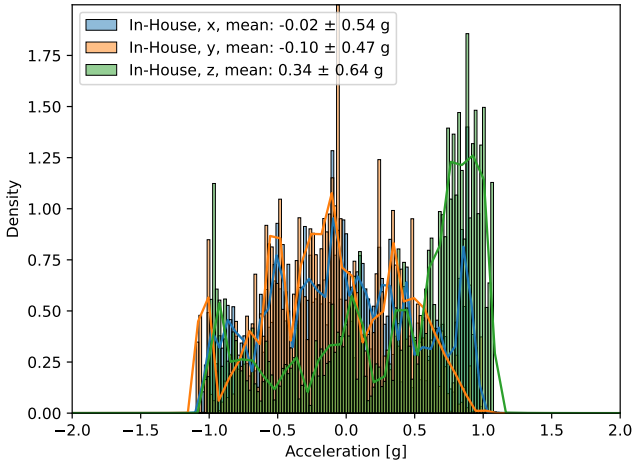

Figure 16: In-House

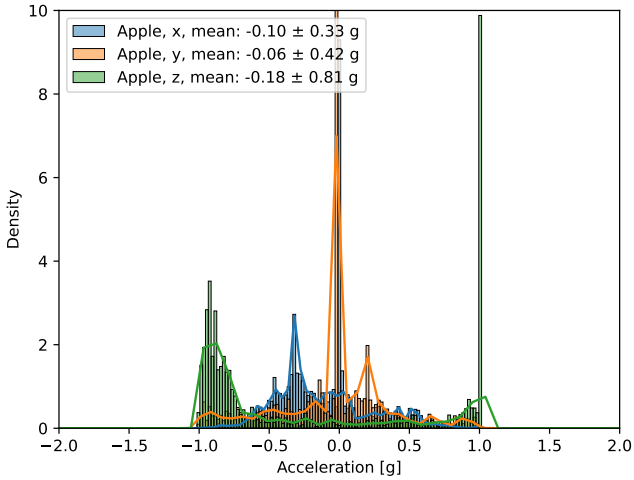

Figure 17: Apple Watch

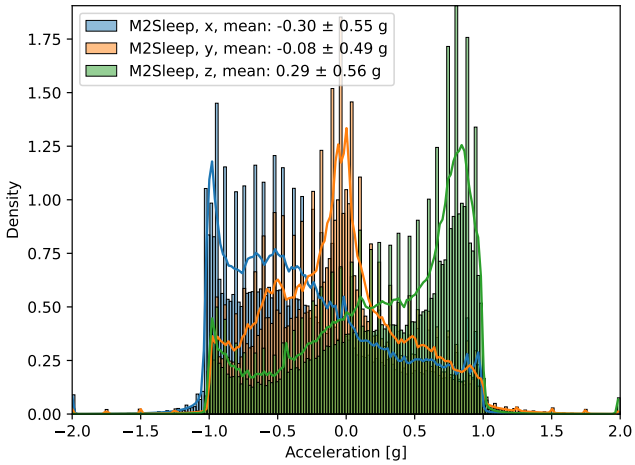

Figure 18: M2Sleep

Figure 19: Distribution of acceleration values for in-house (16), Apple Watch (17), and M2Sleep (18) dataset in units of g ($9.81 \frac{m}{s^2}$).

Model	Learning rate	Batch Size	Optimizer	Weight Decay	τ	EMA	M	Epoch
BYOL	1e-3	64	Adam	1.5e-6	-	0.996	-	60
SimSiam	3e-4	256	Adam	1e-4	-	-	-	60
SimCLR	2.5e-3	256	Adam	1e-6	0.1	-	-	120
NNCLR	2e-3	256	Adam	1e-6	0.1	-	1024	120
TS-TCC	3e-4	128	Adam	3e-4	0.2	-	-	40

Table 14: SSL - Hyperparameters for SSL-experiments. Taken from Qian et al. [77], on SHAR dataset.

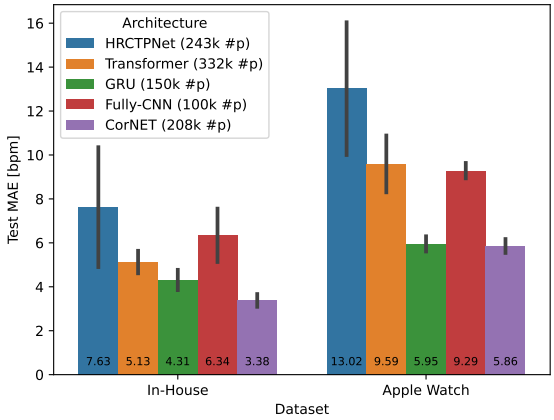

Figure 20: Comparison of mean absolute error (MAE) in heart rate estimation across different neural network architectures using In-House and Apple Watch datasets.

Model Uncertainty	Apple Watch	In-House
Baseline	2.51 ± 0.12	**2.60 ± 0.27**
Classification	2.44 ± 0.05	2.70 ± 0.34
Ensemble	**2.40 ± 0.05**	2.62 ± 0.28
MC Dropout	2.53 ± 0.20	2.84 ± 0.13
NLL	2.56 ± 0.13	2.74 ± 0.26
BNN	2.81 ± 0.23	3.00 ± 0.19

Table 16: Uncertainty - Test MAE [bpm] for different uncertainty models on the In-House and Apple Watch dataset. In both datasets, a single noisy subject was removed from the test set, resulting in lower MAE.

Hyperparameter	Apple Watch	In-House	Baseline Parameters
Framework	SimCLR	NNCLR	NNCLR
Number of pre-training samples	20%	0%	20%
Number of dense layers	1	1	1
Middle layers learning rate	$1 \cdot 10^{-5}$	$5 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
First layers learning rate	$1 \cdot 10^{-5}$	$1 \cdot 10^{-4}$	128
Augmentation 1	Time Warp	-	Time Warp
Augmentation 2	BioInsights	-	BioInsights

Table 15: SSL - Optimal hyperparameters for SSL model in Apple Watch and In-House datasets. The third column contains parameters for downstream experiments.

Dataset	Apple Watch		In-House	
Channel	MAD	MAE	MAD	MAE
X ↔ Y	1.25 ± 2.0	3.31 ± 4.0	1.13 ± 1.1	3.63 ± 3.9
Y ↔ Z	1.34 ± 2.3	3.35 ± 4.1	1.17 ± 1.2	3.68 ± 3.8
Z ↔ X	1.42 ± 2.3	3.31 ± 4.1	1.29 ± 1.4	3.70 ± 3.9
X = 0	2.49 ± 3.8	4.19 ± 5.0	1.89 ± 2.0	4.34 ± 4.3
Y = 0	3.45 ± 5.1	5.15 ± 6.1	1.73 ± 1.8	4.15 ± 4.1
Z = 0	2.76 ± 4.1	4.50 ± 5.4	1.66 ± 1.8	4.14 ± 4.1

Table 17: Follow-Up - MAE on Apple Watch and in-house dataset for channel permutation (upper) and channel occlusion (lower) using the Attention CorNET. The metrics represent the mean amplitude deviation (MAD) from the original prediction and the MAE, both in bpm.

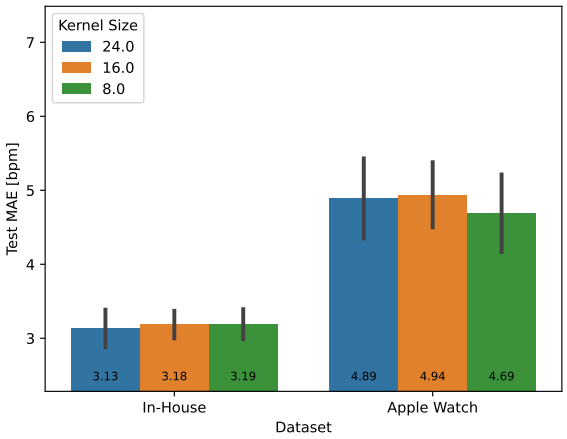

Figure 21: Kernel size

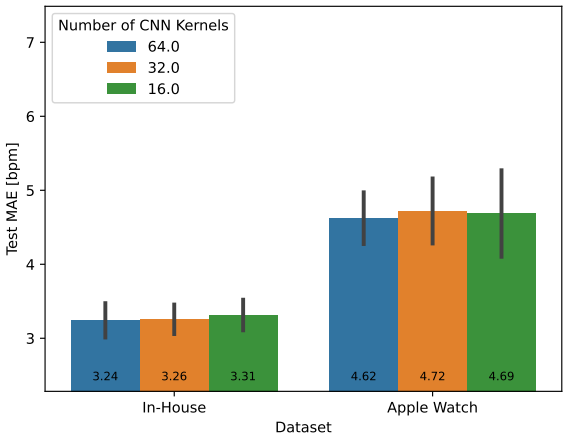

Figure 22: Number of convolutional kernels

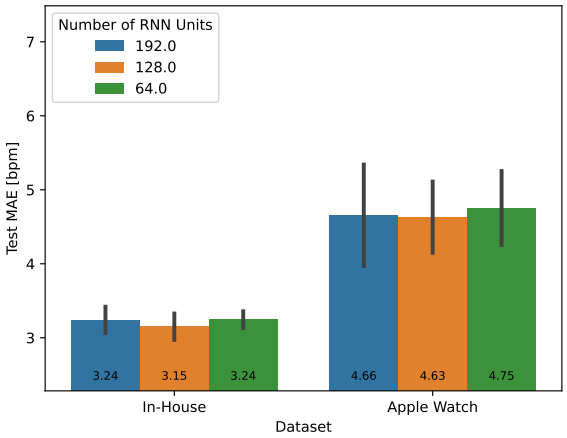

Figure 23: Number of GRU units

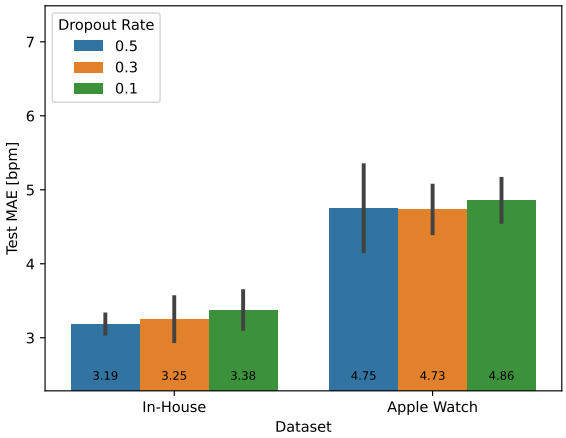

Figure 24: Dropout rate

Figure 25: Baseline - Comparison of mean absolute error (MAE) in heart rate estimation across different kernel sizes (21), number of kernels (22), number of GRU units (23), and dropout rate (24) for CorNET architecture using in-house and Apple Watch datasets

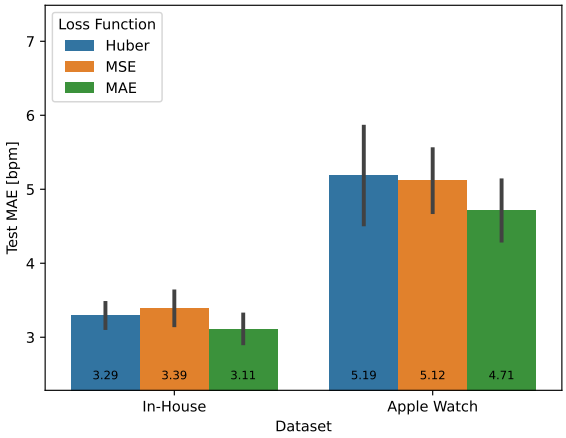

Figure 26: Loss Function; MAE

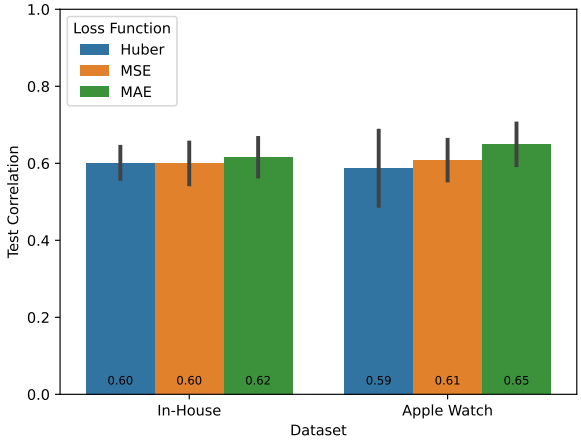

Figure 27: Loss function, Correlation

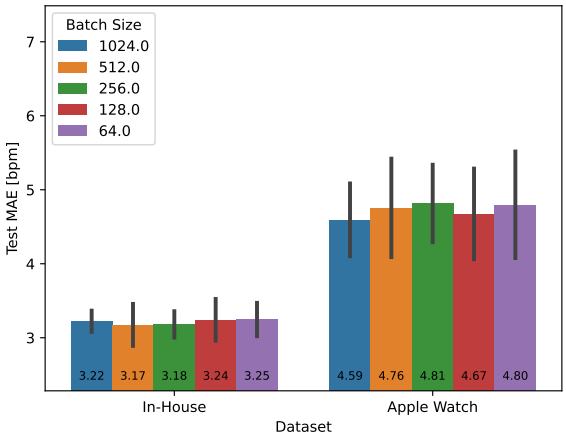

Figure 28: Batch size

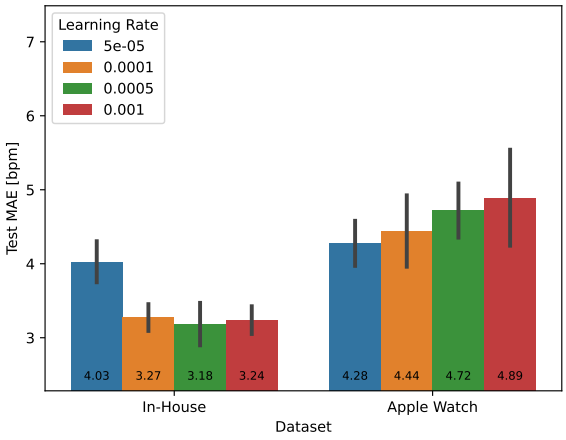

Figure 29: Learning rate

Figure 30: Baseline - Comparison of mean absolute error (MAE) in heart rate estimation across different loss functions (26, 27), batch sizes (28), and learning rates (29) for CorNET architecture using In-House and Apple Watch datasets

	Comparison Model		Baseline Model		Advanced Deep Learning Model				
Subject	Subject Median	Median	Signal Processing	Deep Learning	Attention	Self-Supervised	Uncertainty	Uncertainty + Postprocessing	Self-Supervised + Uncertainty + Postprocessing
2	0	0	0.33	0.47 ± 0.06	0.47 ± 0.04	0.42 ± 0.02	0.47 ± 0.05	**0.52 ± 0.04**	0.49 ± 0.01
6	0	0	0.23	0.53 ± 0.07	0.52 ± 0.03	0.33 ± 0.12	0.52 ± 0.07	**0.56 ± 0.08**	0.42 ± 0.14
15	0	0	0.15	0.57 ± 0.04	0.54 ± 0.03	0.51 ± 0.03	0.56 ± 0.04	**0.66 ± 0.05**	0.64 ± 0.03
20	0	0	0.06	0.12 ± 0.05	0.15 ± 0.03	0.21 ± 0.06	0.12 ± 0.04	0.23 ± 0.06	**0.31 ± 0.08**
In-House (Test)	0.65	0	0.29	0.60 ± 0.05	0.61 ± 0.04	0.57 ± 0.03	0.62 ± 0.04	**0.70 ± 0.04**	0.67 ± 0.02
In-House (Val)	-	-	-	0.56 ± 0.09	0.58 ± 0.09	0.54 ± 0.12	**0.60 ± 0.09**	0.57 ± 0.09	0.57 ± 0.10
46343	0	0	0.56	0.72 ± 0.01	0.73 ± 0.02	0.68 ± 0.03	0.74 ± 0.01	**0.80 ± 0.01**	0.74 ± 0.05
844359	0	0	0.66	0.81 ± 0.01	0.82 ± 0.01	0.79 ± 0.01	0.82 ± 0.01	**0.86 ± 0.01**	0.83 ± 0.01
4018081	0	0	0.49	0.61 ± 0.03	0.63 ± 0.02	0.60 ± 0.03	0.63 ± 0.02	0.75 ± 0.03	**0.78 ± 0.01**
8258170	0	0	0.09	0.11 ± 0.08	0.03 ± 0.10	0.23 ± 0.06	0.10 ± 0.10	0.24 ± 0.16	**0.36 ± 0.15**
9618981	0	0	0.31	0.47 ± 0.01	0.49 ± 0.01	0.42 ± 0.01	0.51 ± 0.01	**0.72 ± 0.02**	0.69 ± 0.02
Apple Watch (Test)	0.83	0	0.71	0.65 ± 0.05	0.63 ± 0.07	0.78 ± 0.03	0.65 ± 0.06	0.74 ± 0.08	**0.85 ± 0.04**
Apple Watch (Val)	-	-	-	0.81 ± 0.06	0.83 ± 0.05	0.79 ± 0.05	**0.85 ± 0.05**	0.84 ± 0.05	0.79 ± 0.06

Table 18: Final results - Comparison of Correlation metrics of different models on the subjects of the Apple Watch Dataset and in-house dataset.

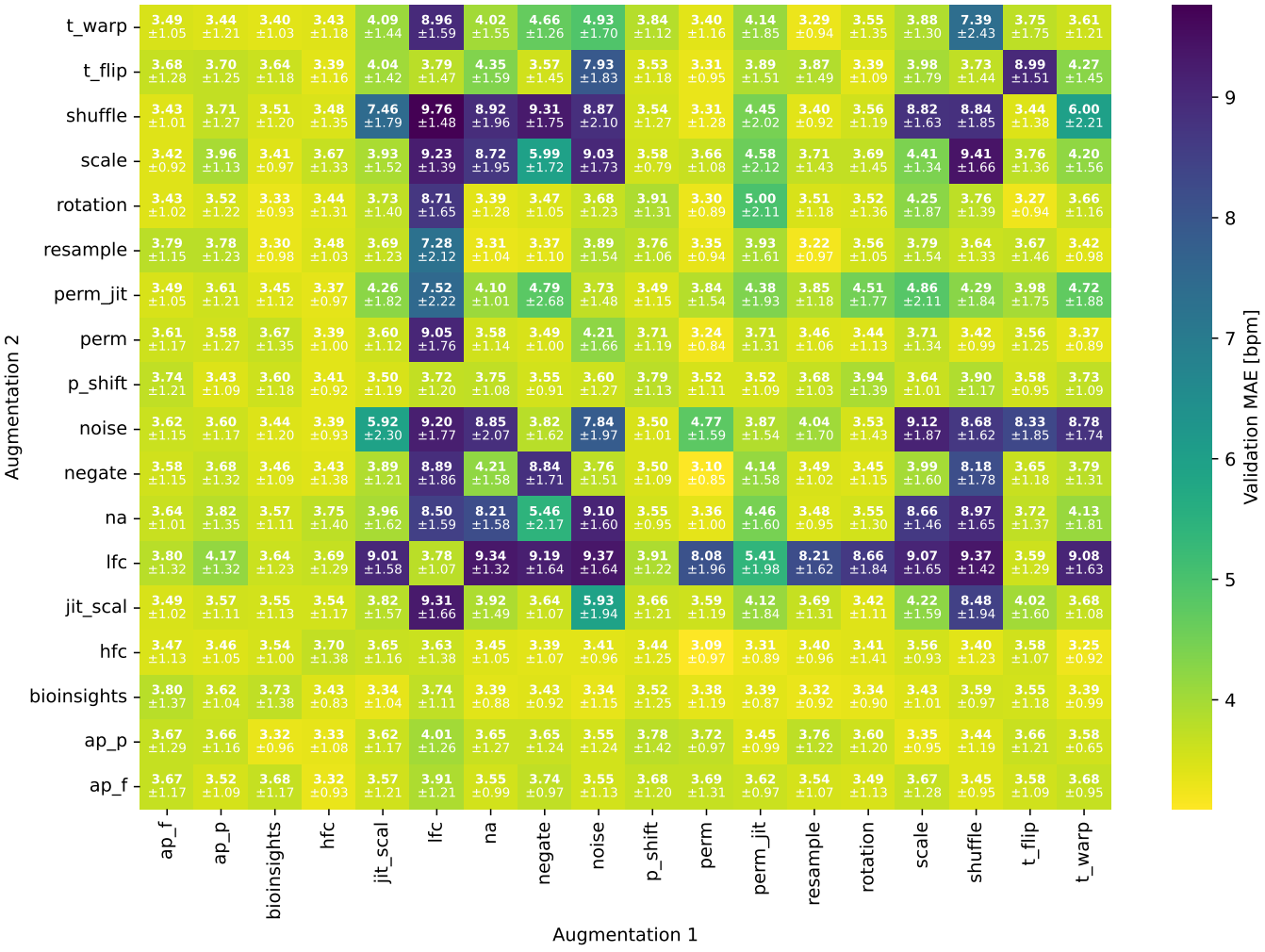

Figure 31: SSL - Comparison of the validation set downstream mean absolute error (MAE) in heart rate estimation for different combinations of augmentations. All models were pre-trained with the *Simsiam* framework and fine-tuned with 5-fold cross-validation on the Apple Watch dataset. The x and y axes contain the names of the augmentations (see Section 3.7). *na* stands for no augmentation.

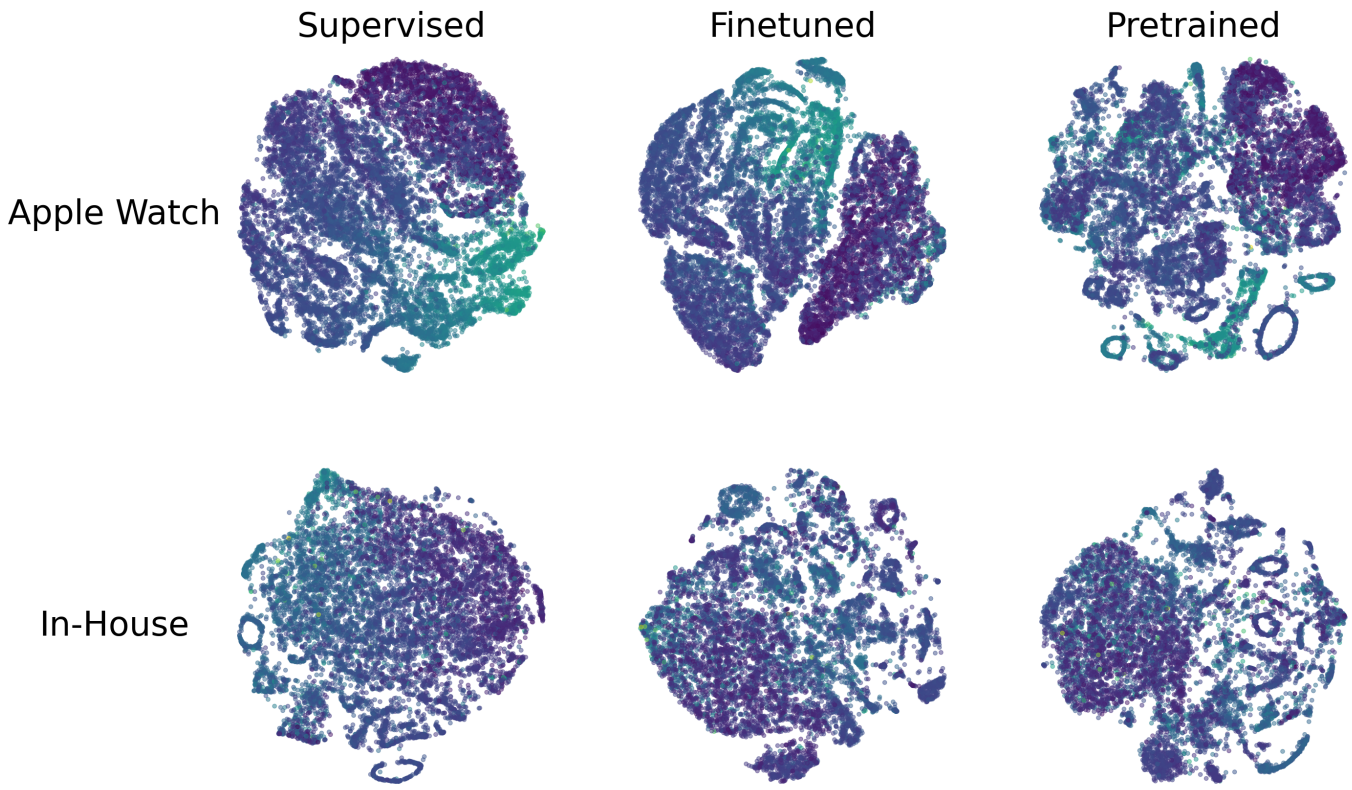

Figure 32: tSNE-plots for extracted features from Apple Watch and in-house dataset. The plots are from models that were trained using supervised learning, SSL pre-training + fine-tuning, and just SSL pre-training. The color encodes the ground truth heart rate.

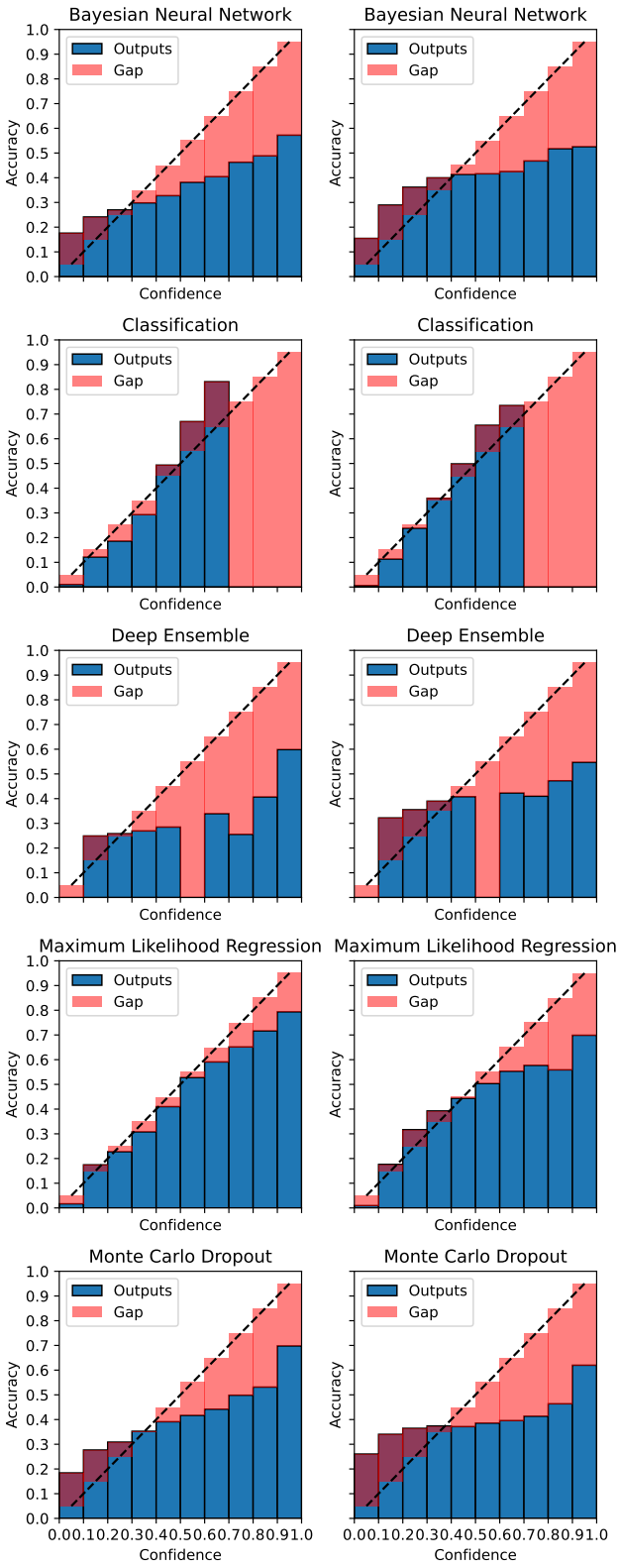

Figure 33: Uncertainty - Reliability Diagram for Apple Watch dataset (left) and In-House dataset (right) for five different uncertainty models. If a confidence bar does not exist, it means that the model did not predict that confidence on the dataset.

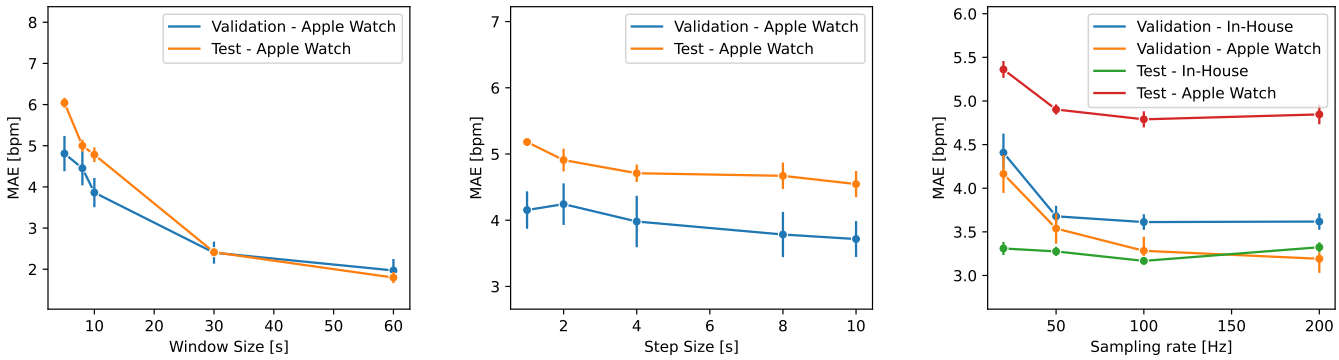

Figure 34: Ablations - Comparison of mean absolute error (MAE) in heart rate estimation across different parameters using In-House and Apple Watch datasets. The evaluated parameters are window size (left), step size (middle), and sampling rate(right). The colors in each Figure encode the dataset and the partition.

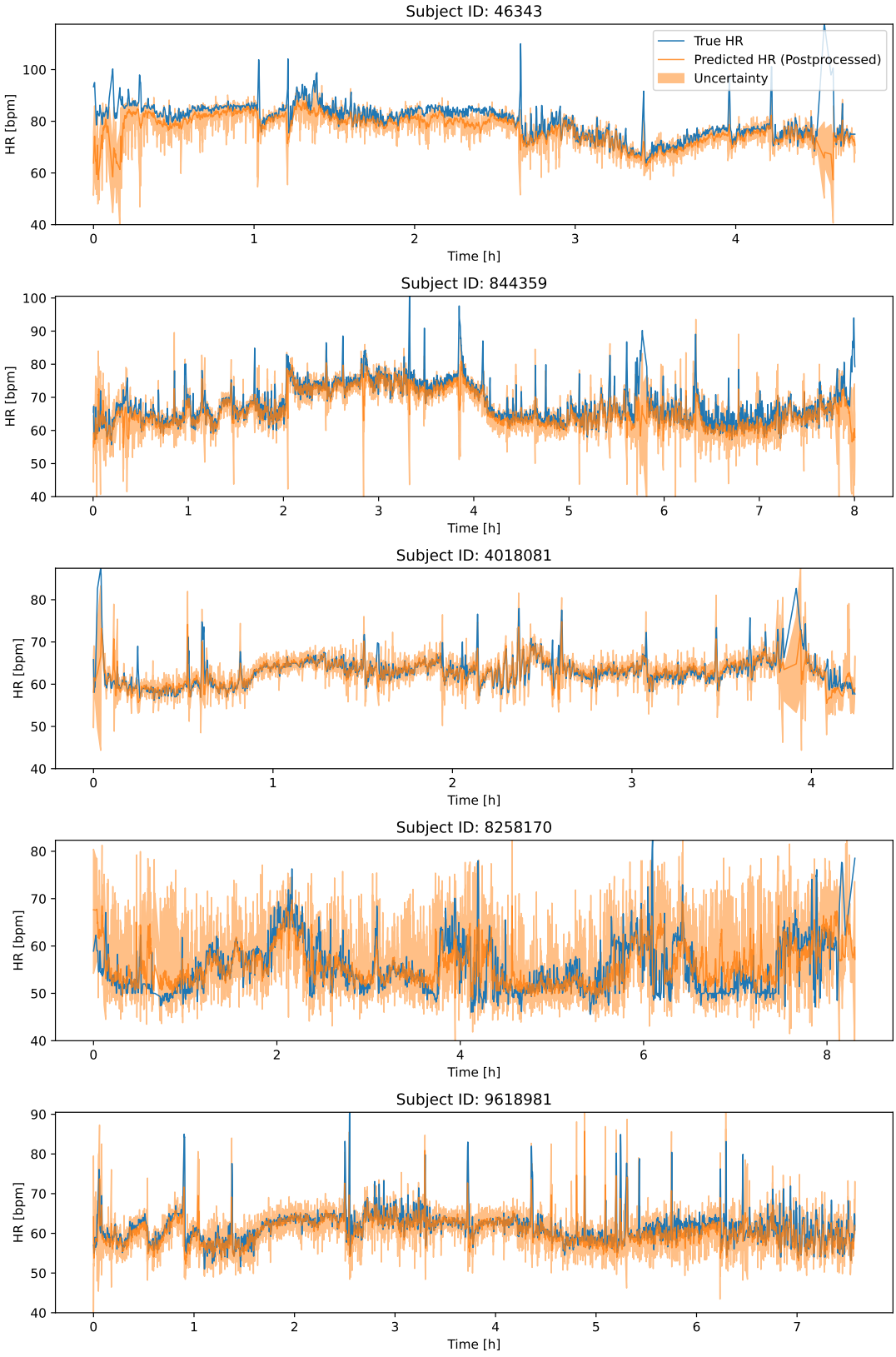

Figure 35: Final Model - Heart Rate and uncertainty predictions on the Apple Watch dataset for all subjects in the test set, as seen in Table 13. The predictions come from the Uncertainty + Postprocessing model.

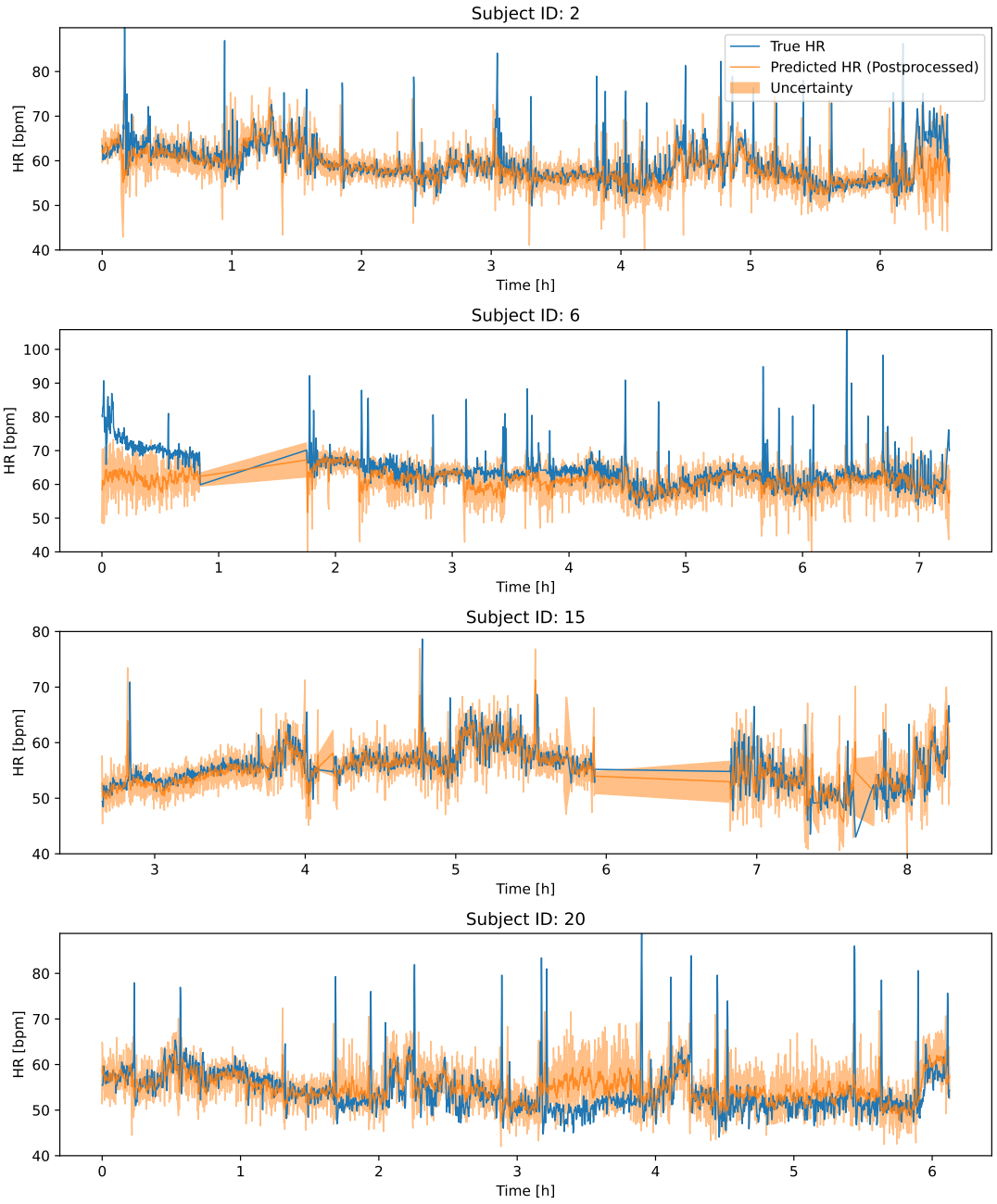

Figure 36: Final results - Comparison of correlation metric of different models on the subjects of the Apple Watch dataset and in-house dataset. The signal processing model is the adapted BioInsights model. The deep learning model is the optimized CorNET model. All advanced models extend the baseline deep learning mode. The attention model adds channel attention. The self-supervised learning model consists of a contrastive NNCLR model, pre-trained on the Capture24 dataset. The uncertainty model uses maximum likelihood regression and the postprocessing model uses belief propagation. Combinations of models use the previously mentioned models. All shown subjects are in the test sets of their respective datasets, except for the summarized subjects from the validation set, marked with (val).

Acronyms

ACC acceleration. 6, 7

BCG Ballistocardiography. 1–9, 11, 13, 14, 17, 25, 29–31

BNN bayesian neural network. 9, 10, 15, 21, 28, 29

CNN convoluational neural network. 13

ECE expected calibration error. 11, 21, 28

ECG Electrocardiography. 1, 3–5, 7, 9, 12, 14, 25, 31

EEMD empirical mode decomposition. 4, 6

ELBO evidence lower bound. 10

ELM extreme learning machine. 5

FFT fast fourier transform. 4, 5, 13, 26

GAN generative adversarial network. 5, 8

GRU gated recurrent unit. 13, 14, 27, 32, 34

HAR human activity recognition. 2, 8, 11, 14, 25, 27, 28

HMM hidden Markov model. 5, 11, 16, 21, 28

HR heart rate. 3, 6, 12, 24, 25, 29

HRV heart rate variability. 31

ICA independent component analysis. 4

IMU inertial measurement unit. 30

KL Kullback-Leibler. 10, 15

LSTM long short-term memory. 13, 14

MA motion artifact. 2–6

MAD mean amplitude deviation. 6, 25, 29, 33

MAE mean absolute error. 1, 2, 4, 6, 7, 12, 14, 15, 17–36, 39

MAP maximum a posteriori. 9, 10

MCMC Markov Chain Monte Carlo. 10

MLE maximum likelihood estimation. 9, 10

MLP multilayer perceptron. 8

MSE mean squared error. 8, 14, 27

PPG Photoplethysmogram. 1–9, 12, 13, 25–28, 30, 31

RNN recurrent neural network. 13–15, 19, 22, 28, 32

SCG Seismocardiography. 3

SSA singular spectrum analysis. 4, 6, 13, 17, 26, 27

SSL self-supervised learning. 1, 2, 8, 9, 14–16, 18–22, 25–31, 33, 36, 37

SSR sparse signal reconstruction. 13

STD standard deviation. 29

STFT short-time fourier transform. 4

VMD variable mode decomposition. 4

WSST wavelet synchrosqueezed transform. 4

REFERENCES

[1] John Allen. 2007. Photoplethysmography and its application in clinical physiological measurement. (3 2007). Issue 3. DOI: http://dx.doi.org/10.1088/0967-3334/28/3/R01

[2] Alexander Amini, Wilko Schwarting, A. Soleimany, and D. Rus. 2019. Deep Evidential Regression. *Neural Information Processing Systems* (2019).

[3] Robert Avram, Geoffrey H. Tison, Kirstin Aschbacher, Peter Kuhar, Eric Vittinghoff, Michael Butzner, Ryan Runge, Nancy Wu, Mark J. Pletcher, Gregory M. Marcus, and Jeffrey Olgin. 2019. Real-world heart rate norms in the Health eHeart study. *NPJ Digital Medicine* 2 (12 2019). Issue 1. DOI: http://dx.doi.org/10.1038/S41746-019-0134-9

[4] Valentin Bieri, Paul Streli, Berken Utku Demirel, and Christian Holz. 2023. BeliefPPG: Uncertainty-aware Heart Rate Estimation from PPG signals via Belief Propagation. (6 2023). http://arxiv.org/abs/2306.07730

[5] Dwaipayan Biswas, Neide Simoes-Capela, Chris Van Hoof, and Nick Van Helleputte. 2019. Heart Rate Estimation from Wrist-Worn Photoplethysmography: A Review. *IEEE Sensors Journal* 19 (8 2019), 6560–6570. Issue 16. DOI: http://dx.doi.org/10.1109/JSEN.2019.2914166

[6] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight Uncertainty in Neural Networks. *32nd International Conference on Machine Learning, ICML 2015* 2 (5 2015), 1613–1622. https://arxiv.org/abs/1505.05424v2

[7] Alessio Burrello, Daniele Jahier Pagliari, Pierangelo Maria Rapa, Matilde Semilia, Matteo Risso, Tommaso Polonelli, Massimo Poncino, Luca Benini, and Simone Benatti. 2022. Embedding Temporal Convolutional Networks for Energy-efficient PPG-based Heart Rate Monitoring. *ACM Transactions on Computing for Healthcare* 3 (4 2022). Issue 2. DOI: http://dx.doi.org/10.1145/3487910

[8] Alessio Burrello, Daniele Jahier Pagliari, Matteo Risso, Simone Benatti, Enrico MacIi, Luca Benini, and Massimo Poncino. 2021. Q-PPG: Energy-Efficient PPG-Based Heart Rate Monitoring on Wearable Devices. *IEEE Transactions on Biomedical Circuits and Systems* 15 (12 2021), 1196–1209. Issue 6. DOI: http://dx.doi.org/10.1109/TBCAS.2021.3122017

[9] Guillaume Cathelain, Bertrand Rivet, Sophie Achard, Jean Bergounioux, and Francois Jouen. 2020. U-Net Neural Network for Heartbeat Detection in Ballistocardiography. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference* 2020 (7 2020), 465–468. DOI: http://dx.doi.org/10.1109/EMBC44109.2020.9176687

[10] Shing Chan, Hang Yuan, Catherine Tong, Aidan Acquah, Abram Schonfeldt, Jonathan Gershuny, and Aiden Doherty. 2024. CAPTURE-24: A large dataset of wrist-worn activity tracker data collected in the wild for human activity recognition. (2 2024). https://arxiv.org/abs/2402.19229v1

[11] Xiangmao Chang, Gangkai Li, Linlin Tu, Guoliang Xing, and Tian Hao. 2019. DeepHeart: Accurate heart rate estimation from PPG signals based on deep learning. *Proceedings - 2019 IEEE 16th International Conference on Mobile Ad Hoc and Smart Systems, MASS 2019* (11 2019), 371–379. DOI: http://dx.doi.org/10.1109/MASS.2019.00051

[12] Xiangmao Chang, Gangkai Li, Guoliang Xing, Kun Zhu, and Linlin Tu. 2021. DeepHeart: A Deep Learning Approach for Accurate Heart Rate Estimation from PPG Signals. *ACM Transactions on Sensor Networks* 17 (6 2021). Issue 2. DOI:http://dx.doi.org/10.1145/3441626

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. *37th International Conference on Machine Learning, ICML 2020* PartF168147-3 (2 2020), 1575–1585. https://arxiv.org/abs/2002.05709v3

[14] Sun-Taag Choe and W. Cho. 2017. Simplified real-time heartbeat detection in ballistocardiography using a dispersion-maximum method. *Biomedical Research-tokyo* (2017).

[15] Karmel W. Choi, Chia Yen Chen, Murray B. Stein, Yann C. Klimentidis, Min Jung Wang, Karestan C. Koenen, Jordan W. Smoller, Naomi R. Wray, Stephan Ripke, Manuel Mattheisen, Maciej Trzaskowski, Enda M. Byrne, Abdel Abdellaoui, Mark J. Adams, Esben Agerbo, Tracy M. Air, Till F.M. Andlauer, Silviu Alin Bacanu, Marie Bækvad-Hansen, Aartjan T.F. Beekman, Tim B. Bigdeli, Elisabeth B. Binder, Douglas H.R. Blackwood, Julien Bryois, Henriette N. Buttenschøn, Jonas Bybjerg-Grauholm, Na Cai, Enrique Castelao, Jane Hvarregaard Christensen, Toni Kim Clarke, Jonathan R.I. Coleman, Lucía Colodro-Conde, Baptiste Couvy-Duchesne, Nick Craddock, Gregory E. Crawford, Gail Davies, Ian J. Deary, Franziska Degenhardt, Eske M. Derks, Nese Direk, Conor V. Dolan, Erin C. Dunn, Thalia C. Eley, Valentina Escott-Price, Farnush Farhadi Hassan Kiadeh, Hilary K. Finucane, Andreas J. Forstner, Josef Frank, Héléna A. Gaspar, Michael Gill, Fernando S. Goes, Scott D. Gordon, Jakob Grove, Lynsey S. Hall, Christine Søholm Hansen, Thomas F. Hansen, Stefan Herms, Ian B. Hickie, Per Hoffmann, Georg Homuth, Carsten Horn, Jouke Jan Hottenga, David M. Hougaard, Marcus Ising, Rick Jansen, Eric Jorgenson, James A. Knowles, Isaac S. Kohane, Julia Kraft, Warren W. Kretzschmar, Jesper Krogh, Zoltán Kutalik, Yihan Li, Penelope A. Lind, Donald J. MacIntyre, Dean F. MacKinnon, Robert M. Maier, Wolfgang Maier, Jonathan Marchini, Hamdi Mbarek, Patrick McGrath, Peter McGuffin, Sarah E. Medland, Divya Mehta, Christel M. Middeldorp, Evelin Mihailov, Yuri Milaneschi, Lili Milani, Francis M. Mondimore, Grant W. Montgomery, Sara Mostafavi, Niamh Mullins, Matthias Nauck, Bernard Ng, Michel G. Nivard, Dale R. Nyholt, Paul F. O'Reilly, Hogni Oskarsson, Michael J. Owen, Jodie N. Painter, Carsten Bøcker Pedersen, Marianne Giørtz Pedersen, Roseann E. Peterson, Erik Pettersson, Wouter J. Peyrot, Giorgio Pistis, Danielle Posthuma, Jorge A. Quiroz, Per Qvist, John P. Rice, Brien P. Riley, Margarita Rivera, Saira Saeed Mirza, Robert Schoevers, Eva C. Schulte, Ling Shen, Jianxin Shi, Stanley I. Shyn, Engilbert Sigurdsson, Grant C.B. Sinnamon, Johannes H. Smit, Daniel J. Smith, Hreinn Stefansson, Stacy Steinberg, Fabian Streit, Jana Strohmaier, Katherine E. Tansey, Henning Teismann, Alexander Teumer, Wesley Thompson, Pippa A. Thomson, Thorgeir E. Thorgeirsson, Matthew Traylor, Jens Treutlein, Vassily Trubetskoy, André G. Uitterlinden, Daniel Umbricht, Sandra van der Auwera, Albert M. van Hemert, Alexander Viktorin, Peter M. Visscher, Yunpeng Wang, Bradley T. Webb, Shantel Marie Weinsheimer, Jürgen Wellmann, Gonneke Willemsen, Stephanie H. Witt, Yang Wu, Hualin S. Xi, Jian Yang, Futao Zhang, Volker Arolt, Bernhard T. Baune, Klaus Berger, Dorret I. Boomsma, Sven Cichon, Udo Dannlowski, E. J.C. de Geus, J. Raymond DePaulo, Enrico Domenici, Katharina Domschke, Tõnu Esko, Hans J. Grabe, Steven P. Hamilton, Caroline Hayward, Andrew C. Heath, Kenneth S. Kendler, Stefan Kloiber, Glyn Lewis, Qingqin S. Li, Susanne Lucae, Pamela A.F. Madden, Patrik K. Magnusson, Nicholas G. Martin, Andrew M. McIntosh, Andres Metspalu, Ole Mors, Preben Bo Mortensen, Bertram Müller-Myhsok, Merete Nordentoft, Markus M. Nöthen, Michael C. O'Donovan, Sara A. Paciga, Nancy L. Pedersen, Brenda W.J.H. Penninx, Roy H. Perlis, David J. Porteous, James B. Potash, Martin Preisig, Marcella Rietschel, Catherine Schaefer, Thomas G. Schulze, Kari Stefansson, Henning Tiemeier, Rudolf Uher, Henry Völzke, Myrna M. Weissman, Thomas Werge, Cathryn M. Lewis, Douglas F. Levinson, Gerome Breen, Anders D. Børglum, and Patrick F. Sullivan. 2019. Assessment of Bidirectional

Relationships Between Physical Activity and Depression Among Adults: A 2-Sample Mendelian Randomization Study. *JAMA Psychiatry* 76 (4 2019), 399–408. Issue 4. DOI: http://dx.doi.org/10.1001/JAMAPSYCHIATRY.2018.4175

[16] Heewon Chung, Hoon Ko, Hooseok Lee, and Jinseok Lee. 2020. Deep Learning for Heart Rate Estimation From Reflectance Photoplethysmography With Acceleration Power Spectrum and Acceleration Intensity. *IEEE Access* 8 (2020), 63390–63402. DOI: http://dx.doi.org/10.1109/ACCESS.2020.2981956

[17] Heewon Chung, Hooseok Lee, and Jinseok Lee. 2019. Finite State Machine Framework for Instantaneous Heart Rate Validation Using Wearable Photoplethysmography during Intensive Exercise. *IEEE Journal of Biomedical and Health Informatics* 23 (7 2019), 1595–1606. Issue 4. DOI:http://dx.doi.org/10.1109/JBHI.2018.2871177

[18] Aiden Doherty, Dan Jackson, Nils Hammerla, Thomas Plötz, Patrick Olivier, Malcolm H. Granat, Tom White, Vincent T. Van Hees, Michael I. Trenell, Christoper G. Owen, Stephen J. Preece, Rob Gillions, Simon Sheard, Tim Peakman, Soren Brage, and Nicholas J. Wareham. 2017. Large Scale Population Assessment of Physical Activity Using Wrist Worn Accelerometers: The UK Biobank Study. *PLOS ONE* 12 (2 2017), e0169649. Issue 2. DOI: http://dx.doi.org/10.1371/JOURNAL.PONE.0169649

[19] Pan Du, Warren A. Kibbe, and Simon M. Lin. 2006. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics* 22 (9 2006), 2059–2065. Issue 17. DOI: http://dx.doi.org/10.1093/BIOINFORMATICS/BTL355

[20] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. 2021. With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations. *Proceedings of the IEEE International Conference on Computer Vision* (4 2021), 9568–9577. DOI: http://dx.doi.org/10.1109/ICCV48922.2021.00945

[21] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. 2021. Time-Series Representation Learning via Temporal and Contextual Contrasting. *IJCAI International Joint Conference on Artificial Intelligence* (6 2021), 2352–2359. DOI: http://dx.doi.org/10.24963/ijcai.2021/324

[22] Mohamed Elgendi. 2016. Optimal Signal Quality Index for Photoplethysmogram Signals. *Bioengineering* 3 (12 2016). Issue 4. DOI: http://dx.doi.org/10.3390/BIOENGINEERING3040021

[23] Yu Fang, Zhongwei Jiang, and Haibin Wang. 2018. A Novel Sleep Respiratory Rate Detection Method for Obstructive Sleep Apnea Based on Characteristic Moment Waveform. *Journal of Healthcare Engineering* 2018 (2018). DOI: http://dx.doi.org/10.1155/2018/1902176

[24] Association for the Advancement of Medical Instrumentation and others. 2002. Cardiac monitors, heart rate meters, and alarms. *American National Standard (ANSI/AAMI EC13: 2002) Arlington, VA* (2002), 1–87.

[25] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. 2019. Deep Ensembles: A Loss Landscape Perspective. (12 2019). https://arxiv.org/abs/1912.02757v2

[26] Kim Fox, Jeffrey S. Borer, A. John Camm, Nicolas Danchin, Roberto Ferrari, Jose L. Lopez Sendon, Philippe Gabriel Steg, Jean Claude Tardif, Luigi Tavazzi, and Michal Tendera. 2007. Resting Heart Rate in Cardiovascular Disease. *Journal of the American College of Cardiology* 50 (8 2007), 823–830. Issue 9. DOI:http://dx.doi.org/10.1016/J.JACC.2007.04.079

[27] David Friedrich, Xavier L. Aubert, Hartmut Führ, and Andreas Brauers. 2010. Heart rate estimation on a beat-to-beat basis via ballistocardiography - a hybrid approach. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology* (2010), 4048–4051. DOI: http://dx.doi.org/10.1109/IEMBS.2010.5627626

[28] Hayato Fukushima, Haruki Kawanaka, Md Shoaib Bhuiyan, and Koji Oguri. 2012. Estimating heart rate using wrist-type Photoplethysmography and acceleration sensor while running. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2012), 2901–2904. DOI: http://dx.doi.org/10.1109/EMBC.2012.6346570

[29] Maksym Gaiduk, Thomas Penzel, Juan Antonio Ortega, and Ralf Seepold. 2018. Automatic sleep stages classification using respiratory, heart rate and movement signals. *Physiological Measurement* 39 (12 2018), 124008. Issue 12. DOI: http://dx.doi.org/10.1088/1361-6579/AAF5D4

[30] Yarin Gal and Zoubin Ghahramani. 2015. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. (6 2015). http://arxiv.org/abs/1506.02142

[31] Shkurta Gashi, Lidia Alecci, Elena Di Lascio, Maike E. Debus, Francesca Gasparini, and Silvia Santini. 2022. The Role of Model Personalization for Sleep Stage and Sleep Quality Recognition Using Wearables. *IEEE Pervasive Computing* 21 (2022), 69–77. Issue 2. DOI: http://dx.doi.org/10.1109/MPRV.2022.3164334

[32] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. 2021. A Survey of Uncertainty in Deep Neural Networks. *Artificial Intelligence Review* 56 (7 2021), 1513–1589. DOI: http://dx.doi.org/10.1007/s10462-023-10562-9

[33] Tomas I. Gonzales, Justin Y. Jeon, Timothy Lindsay, Kate Westgate, Ignacio Perez-Pozuelo, Stefanie Hollidge, Katrien Wijndaele, Kirsten Rennie, Nita Forouhi, Simon Griffin, Nick Wareham, and Soren Brage. 2023. Resting heart rate is a population-level biomarker of cardiorespiratory fitness: The Fenland Study. *PLoS ONE* 18 (5 2023). Issue 5 May. DOI: http://dx.doi.org/10.1371/journal.pone.0285272

[34] Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings* (12 2013). https://arxiv.org/abs/1312.6211v3

[35] By J W Gordon. 1877. Certain Molar Movements of the Human Body produced by the Circulation of the Blood. *Journal of Anatomy and Physiology* 11 (4 1877), 533. Issue Pt 3. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1309740/

[36] Irina F. Gorodnitsky and Bhaskar D. Rao. 1997. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing* 45 (1997), 600–616. Issue 3. DOI:http://dx.doi.org/10.1109/78.558475

[37] Jean Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap your own latent: A new approach to self-supervised Learning. *Advances in Neural Information Processing Systems* 2020-December (6 2020). https://arxiv.org/abs/2006.07733v3

[38] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. *34th International Conference on Machine Learning, ICML 2017* 3 (6 2017), 2130–2143. https://arxiv.org/abs/1706.04599v2

[39] Marian Haescher, Denys J.C. Matthies, John Trimpop, and Bodo Urban. 2015. A study on measuring heart- and respiration-rate via wrist-worn accelerometer-based seismocardiography (SCG) in Comparison to commonly applied technologies. *ACM International Conference Proceeding Series* 25-26-June-2015 (6 2015). DOI: http://dx.doi.org/10.1145/2790044.2790054

[40] Harish Haresamudram, Apoorva Beedu, Varun Agrawal, Patrick L. Grady, Irfan Essa, Judy Hoffman, and Thomas Plötz. 2020. Masked reconstruction based self-supervision for human activity recognition. *Proceedings - International Symposium on Wearable Computers, ISWC* (9 2020), 45–49. DOI: http://dx.doi.org/10.1145/3410531.3414306

[41] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2021. Contrastive Predictive Coding for Human Activity Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5 (6 2021). Issue 2. DOI:http://dx.doi.org/10.1145/3463506

[42] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2022. Assessing the State of Self-Supervised Human Activity Recognition Using Wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6 (9 2022). Issue 3. DOI: http://dx.doi.org/10.1145/3550299

[43] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. 2019. Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty. *Advances in Neural Information Processing Systems* 32 (2019). https://github.com/hendrycks/ss-ood.

[44] Javier Hernandez, Yin Li, James M. Rehg, and Rosalind W. Picard. 2015a. BioGlass: Physiological parameter estimation using a head-mounted wearable device. *Proceedings of the 2014 4th International Conference on Wireless Mobile Communication and Healthcare - "Transforming Healthcare Through Innovations in Mobile and Wireless Technologies", MOBIHEALTH 2014* (1 2015), 55–58. DOI: http://dx.doi.org/10.1109/MOBIHEALTH.2014.7015908

[45] Javier Hernandez, Daniel McDuff, and Rosalind W. Picard. 2015b. Biowatch: Estimation of heart and breathing rates from wrist motions. *Proceedings of the 2015 9th International Conference on Pervasive Computing Technologies for Healthcare, PervasiveHealth 2015* (12 2015), 169–176. DOI:http://dx.doi.org/10.4108/ICST.PERVASIVEHEALTH.2015.259064

[46] Javier Hernandez, Daniel J. McDuff, and Rosalind W. Picard. 2015c. BioInsights: Extracting personal data from 'Still' wearable motion sensors. *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks, BSN 2015* (10 2015). DOI: http://dx.doi.org/10.1109/BSN.2015.7299354

[47] Emily J. Herron, Steven R. Young, and Thomas E. Potok. 2020. Ensembles of Networks Produced from Neural Architecture Search. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 12321 LNCS (2020), 223–234. DOI:http://dx.doi.org/10.1007/978-3-030-59851-8_14/FIGURES/2

[48] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. 2017. Snapshot Ensembles: Train 1, get M for free. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (4 2017). https://arxiv.org/abs/1704.00109v1

[49] Weiran Huang, Mingyang Yi, Xuyang Zhao, and Zihao Jiang. 2021. Towards the Generalization of Contrastive Self-Supervised Learning. (11 2021). https://arxiv.org/abs/2111.00743v4

[50] Omer T. Inan, Pierre Francois Migeotte, Kwang Suk Park, Mozziyar Etemadi, Kouhyar Tavakolian, Ramon Casanella, John Zanetti, Jens Tank, Irina Funtova, G. Kim Prisk, and Marco Di Rienzo. 2015. Ballistocardiography and Seismocardiography: A Review of Recent Advances. *IEEE Journal of Biomedical and Health Informatics* 19 (7 2015), 1414–1427. Issue 4. DOI: http://dx.doi.org/10.1109/JBHI.2014.2361732

[51] Yash Jain, Chi Ian Tang, Chulhong Min, Fahim Kawsar, and Akhil Mathur. 2022. ColloSSL: Collaborative Self-Supervised Learning for Human Activity Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6 (2 2022). Issue 1. DOI:http://dx.doi.org/10.1145/3517246

[52] Alex Kendall and Yarin Gal. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *Advances in Neural Information Processing Systems* 2017-December (3 2017), 5575–5585. https://arxiv.org/abs/1703.04977v2

[53] Emroz Khan, Forsad Al Hossain, Shiekh Zia Uddin, S. Kaisar Alam, and Md Kamrul Hasan. 2016. A Robust Heart Rate Monitoring Scheme Using Photoplethysmographic Signals Corrupted by Intense Motion Artifacts. *IEEE Transactions on Biomedical Engineering* 63 (3 2016), 550–562. Issue 3. DOI: http://dx.doi.org/10.1109/TBME.2015.2466075

[54] Byung S. Kim and Sun K. Yoo. 2006. Motion artifact reduction in photoplethysmography using independent component analysis. *IEEE Transactions on Biomedical Engineering* 53 (3 2006), 566–568. Issue 3. DOI: http://dx.doi.org/10.1109/TBME.2005.869784

[55] Juha M. Kortelainen, Martin O. Mendez, Anna Maria Bianchi, Matteo Matteucci, and Sergio Cerutti. 2010. Sleep staging based on signals acquired through bed sensor. *IEEE Transactions on Information Technology in Biomedicine* 14 (5 2010), 776–785. Issue 3. DOI: http://dx.doi.org/10.1109/TITB.2010.2044797

[56] Rajet Krishnan, Balasubramaniam Natarajan, and Steve Warren. 2010. Two-stage approach for detection and reduction of motion artifacts in photoplethysmographic data. *IEEE Transactions on Biomedical Engineering* 57 (2010), 1867–1876. Issue 8. DOI: http://dx.doi.org/10.1109/TBME.2009.2039568

[57] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. 2018. Accurate Uncertainties for Deep Learning Using Calibrated Regression. *35th International Conference on Machine Learning, ICML 2018* 6 (7 2018), 4369–4377. https://arxiv.org/abs/1807.00263v1

[58] Benjamin Lam, Michael Catt, Sophie Cassidy, Jaume Bacardit, Philip Darke, Sam Butterfield, Ossama Alshabrawy, Michael Trenell, and Paolo Missier. 2021. Using Wearable Activity Trackers to Predict Type 2 Diabetes: Machine Learning-Based Cross-sectional Study of the UK Biobank Accelerometer Cohort. *JMIR diabetes* 6 (3 2021), e23364. Issue 1. DOI: http://dx.doi.org/10.2196/23364

[59] Boreom Lee, Jonghee Han, Hyun Jae Baek, Jae Hyuk Shin, Kwang Suk Park, and Won Jin Yi. 2010. Improved elimination of motion artifacts from a photoplethysmographic signal using a Kalman smoother with simultaneous accelerometry. *Physiological Measurement* 31 (2010), 1585–1603. Issue 12. DOI: http://dx.doi.org/10.1088/0967-3334/31/12/003

[60] Jin Li. 2017. Assessing the accuracy of predictive models for numerical data: Not r nor r2, why not? Then what? *PLoS ONE* 12 (8 2017). Issue 8. DOI: http://dx.doi.org/10.1371/JOURNAL.PONE.0183250

[61] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020. Self-supervised Learning: Generative or Contrastive. *IEEE Transactions on Knowledge and Data Engineering* 35 (6 2020), 857–876. Issue 1. DOI: http://dx.doi.org/10.1109/TKDE.2021.3090866

[62] Ioannis E. Livieris, Lazaros Iliadis, and Panagiotis Pintelas. 2021. On ensemble techniques of weight-constrained neural networks. *Evolving Systems* 12 (3 2021), 155–167. Issue 1. DOI:http://dx.doi.org/10.1007/S12530-019-09324-2/FIGURES/6

[63] Han Lu, Haihong Zhang, Zhiping Lin, and Ng Soon Huat. 2018. *A Novel Deep Learning based Neural Network for Heartbeat Detection in Ballistocardiograph*. DOI:http://dx.doi.org/10.0/Linux-x86_64

[64] David C. Mack, James T. Patrie, Paul M. Suratt, Robin A. Felder, and Majd Alwan. 2009. Development and Preliminary Validation of Heart Rate and Breathing Rate Detection Using a Passive, Ballistocardiography-Based Sleep Monitoring System. *IEEE Transactions on Information Technology in Biomedicine* 13 (2009), 111–120. Issue 1. DOI: http://dx.doi.org/10.1109/TITB.2008.2007194

[65] Yaozong Mai, Zizhao Chen, Baoxian Yu, Ye Li, Zhiqiang Pang, and Zhang Han. 2022. Non-Contact Heartbeat Detection Based on Ballistocardiogram Using UNet and Bidirectional Long Short-Term Memory. *IEEE Journal of Biomedical and Health Informatics* 26 (8 2022), 3720–3730. Issue 8. DOI: http://dx.doi.org/10.1109/JBHI.2022.3162396

[66] Dominique Makowski, Tam Pham, Zen J. Lau, Jan C. Brammer, François Lespinasse, Hung Pham, Christopher Schölzel, and S. H.Annabel Chen. 2021. NeuroKit2: A Python toolbox for neurophysiological signal processing. *Behavior Research Methods* 53 (8 2021), 1689–1696. Issue 4. DOI: http://dx.doi.org/10.3758/S13428-020-01516-Y

[67] Mahdi Boloursaz Mashhadi, Ehsan Asadi, Mohsen Eskandari, Shahrzad Kiani, and Farokh Marvasti. 2015. Heart Rate Tracking using Wrist-Type Photoplethysmographic (PPG) Signals during Physical Exercise with Simultaneous Accelerometry. *IEEE Signal Processing Letters* 23 (12 2015), 227–231. Issue 2. DOI: http://dx.doi.org/10.1109/lsp.2015.2509868

[68] Ryan Mcconville, Ian Craddock, Phillips Netherlands, Robert Piechocki, James Pope, Raul Santos-Rodriguez, Gareth Archer, and Herman Ter Horst. 2018. Online Heart Rate Prediction using Acceleration from a Wrist Worn Wearable. (6 2018). DOI: http://dx.doi.org/doi.org/10.48550/arXiv.1807.04667

[69] Navaneet K. Lakshminarasimha Murthy, Pavan C. Madhusudana, Pradyumna Suresha, Vijitha Periyasamy, and Prasanta Kumar Ghosh. 2015. Multiple Spectral Peak Tracking for Heart Rate Monitoring from Photoplethysmography Signal During Intensive Physical Exercise. *IEEE Signal Processing Letters* 22 (12 2015), 2391–2395. Issue 12. DOI: http://dx.doi.org/10.1109/LSP.2015.2486681

[70] Deebul S Nair, Nico Hochgeschwender, and Miguel A Olivares-Mendez. 2022. Maximum Likelihood Uncertainty Estimation: Robustness to Outliers. (2022).

[71] Loris Nanni, Sheryl Brahnam, and Gianluca Maguolo. 2019. Data Augmentation for Building an Ensemble of Convolutional Neural Networks. *Smart Innovation, Systems and Technologies* 145 (2019), 61–69. DOI:http://dx.doi.org/10.1007/978-981-13-8566-7_6/TABLES/2

[72] Jeremy Nixon, Mike Dusenberry Google, Brain Ghassen, Jerfel Google, Brain Timothy Nguyen, Google Research, Jeremiah Liu, Linchuan Zhang, and Dustin Tran Google Brain. 2019. Measuring Calibration in Deep Learning. (4 2019). https://arxiv.org/abs/1904.01685v2

[73] Christina Orphanidou. 2018. Quality Assessment for the Photoplethysmogram (PPG). (2018), 41–63. DOI: http://dx.doi.org/10.1007/978-3-319-68415-4_3

[74] Huijie Pan, Dogancan Temel, and Ghassan Alregib. 2016. HeartBEAT: Heart Beat Estimation through Adaptive Tracking. (2016), 587–590. DOI: http://dx.doi.org/10.1109/BHI.2016.7455966

[75] Eduardo Pinheiro, Octavian Postolache, and Pedro Girão. 2010. Theory and Developments in an Unobtrusive Cardiovascular System Representation: Ballistocardiography. (2010).

[76] Jun Qi, Jun Du, Sabato Marco Siniscalchi, Xiaoli Ma, and Chin Hui Lee. 2020. On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression. *IEEE Signal Processing Letters* 27 (2020), 1485–1489. DOI: http://dx.doi.org/10.1109/LSP.2020.3016837

[77] Hangwei Qian, Tian Tian, and Chunyan Miao. 2022. What Makes Good Contrastive Learning on Small-Scale Wearable-based Tasks? *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (8 2022), 3761–3771. DOI: http://dx.doi.org/10.1145/3534678.3539134

[78] Daniel Ray, Tim Collins, and Prasad V.S. Ponnapalli. 2022. DeepPulse: An Uncertainty-aware Deep Neural Network for Heart Rate Estimations from Wrist-worn Photoplethysmography. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* 2022-July (2022), 1651–1654. DOI: http://dx.doi.org/10.1109/EMBC48229.2022.9871813

[79] Attila Reiss, Ina Indlekofer, Philip Schmidt, and Kristof Van Laerhoven. 2019. Deep PPG: Large-scale heart rate estimation with convolutional neural networks. *Sensors (Switzerland)* 19 (7 2019). Issue 14. DOI: http://dx.doi.org/10.3390/s19143079

[80] Leandro Giacomini Rocha, Dwaipayan Biswas, Bram Ernst Verhoef, Sergio Bampi, Chris Van Hoof, Mario Konijnenburg, Marian Verhelst, and Nick Van Helleputte. 2020. Binary CorNET: Accelerator for HR Estimation from Wrist-PPG. *IEEE Transactions on Biomedical Circuits and Systems* 14 (8 2020), 715–726. Issue 4. DOI: http://dx.doi.org/10.1109/TBCAS.2020.3001675

[81] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-task Self-Supervised Learning for Human Activity Detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3 (7 2019), 1–30. Issue 2. DOI: http://dx.doi.org/10.1145/3328932

[82] Seyed M.A. Salehizadeh, Duy Dao, Jeffrey Bolkhovsky, Chae Cho, Yitzhak Mendelson, and Ki H. Chon. 2016. A novel time-varying spectral filtering algorithm for reconstruction of motion artifact corrupted heart rate signals during intense physical activities using a wearable photoplethysmogram sensor. *Sensors* 16 (1 2016). Issue 1. DOI: http://dx.doi.org/10.3390/s16010010

[83] D. M. Salerno and J. Zanetti. 1991. Seismocardiography for Monitoring Changes in Left Ventricular Function during Ischemia. *Chest* 100 (10 1991), 991–993. Issue 4. DOI:http://dx.doi.org/10.1378/CHEST.100.4.991

[84] Pritam Sarkar and Ali Etemad. 2020. CardioGAN: Attentive Generative Adversarial Network with Dual Discriminators for Synthesis of ECG from PPG. *35th AAAI Conference on Artificial Intelligence, AAAI 2021* 1 (9 2020), 488–496. DOI: http://dx.doi.org/10.1609/aaai.v35i1.16126

[85] William R. Scarborough, Samuel A. Talbot, John R. Braunstein, Maurice B. Rappaport, William Dock, William F. Hamilton, John E. Smith, John L. Nickerson, and Isaac Starr. 1956. Proposals for Ballistocardiographic Nomenclature and Conventions: Revised and Extended Report of Committee on Ballistocardiographic Terminology. *Circulation* 14 (1956), 435–450. https://api.semanticscholar.org/CorpusID:10327186

[86] Rainer Schubert, Christian Kolbitsch, Stefan Hofbauer, Elias Tappeiner, Karl D. Fritscher, and Samuel M. Pröll. 2021. Heart rate estimation from ballistocardiographic signals using deep learning. *Physiological Measurement* 42 (7 2021), 075005. Issue 7. DOI: http://dx.doi.org/10.1088/1361-6579/AC10AA

[87] Murat Sensoy, Lance Kaplan, and Melih Kandemir. 2018. Evidential Deep Learning to Quantify Classification Uncertainty. *Neural Information Processing Systems* (2018).

[88] A. Shyam, Vignesh Ravichandran, S. P. Preejith, Jayaraj Joseph, and Mohanasankar Sivaprakasam. 2019. PPGnet: Deep Network for Device Independent Heart Rate Estimation from Photoplethysmogram. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (7 2019), 1899–1902. DOI: http://dx.doi.org/10.1109/EMBC.2019.8856989

[89] Seok Bin Song, Jung Woo Nam, and Jin Heon Kim. 2021. Nas-ppg: Ppg-based heart rate estimation using neural architecture search. *IEEE Sensors Journal* 21 (7 2021), 14941–14949. Issue 13. DOI: http://dx.doi.org/10.1109/JSEN.2021.3073047

[90] Dimitris Spathis, Ignacio Perez-Pozuelo, Soren Brage, Nicholas J. Wareham, and Cecilia Mascolo. 2020. Self-supervised transfer learning of physiological representations from free-living wearable data. (11 2020). DOI:http://dx.doi.org/10.1145/3450439.3451863

[91] Isaac Starr, A. J. Rawson, H. A. Schroeder, and N. R. Joseph. 1939. STUDIES ON THE ESTIMATION OF CARDIAC OUPTUT IN MAN, AND OF ABNORMALITIES IN CARDIAC FUNCTION, FROM THE HEART'S RECOIL AND THE BLOOD'S IMPACTS; THE BALLISTOCARDIOGRAM. *https://doi.org/10.1152/ajplegacy.1939.127.1.1* 127 (7 1939), 1–28. Issue 1. DOI: http://dx.doi.org/10.1152/AJPLEGACY.1939.127.1.1

[92] Isaac Starr and Henry A. Schroeder. 1940. BALLISTOCARDIOGRAM. II. NORMAL STANDARDS, ABNORMALITIES COMMONLY FOUND IN DISEASES OF THE HEART AND CIRCULATION, AND THEIR SIGNIFICANCE 1. *Journal of Clinical Investigation* 19 (5 1940), 437–450. Issue 3. DOI:http://dx.doi.org/10.1172/jci101145

[93] ISAAC STARR and FRANCIS C. WOOD. 1961. Twenty-Year Studies with the Ballistocardiograph: The Relation between the Amplitude of the First Record of. *Circulation* 23 (5 1961), 714–732. Issue 5. DOI: http://dx.doi.org/10.1161/01.CIR.23.5.714

[94] Torjus L. Steffensen, Filip E. Schjerven, Hans M. Flade, Idar Kirkeby-Garstad, Emma Ingeström, Fredrik S. Solberg, and Martin Steinert. 2023. Wrist ballistocardiography and invasively recorded blood pressure in healthy volunteers during reclining bike exercise. *Frontiers in Physiology* 14 (2023). DOI: http://dx.doi.org/10.3389/fphys.2023.1189732

[95] Oliver Stegle, Sebastian V. Fallert, David J.C. MacKay, and Søren Brage. 2008. Gaussian process robust regression for noisy heart rate data. *IEEE Transactions on Biomedical Engineering* 55 (9 2008), 2143–2151. Issue 9. DOI: http://dx.doi.org/10.1109/TBME.2008.923118

[96] Paul Streli, Jiaxi Jiang, Andreas Rene Fender, Manuel Meier, Hugo Romat, and Christian Holz. 2022. TapType: Ten-finger text entry on everyday surfaces via Bayesian inference. *Conference on Human Factors in Computing Systems - Proceedings* (4 2022). DOI: http://dx.doi.org/10.1145/3491102.3501878

[97] Xuxue Sun, Ping Yang, Yulin Li, Zhifan Gao, and Yuan Ting Zhang. 2012. Robust heart beat detection from photoplethysmography interlaced with motion artifacts based on empirical mode decomposition. *Proceedings - IEEE-EMBS International Conference on Biomedical and Health Informatics: Global Grand Challenge of Health Informatics, BHI 2012* (2012), 775–778. DOI: http://dx.doi.org/10.1109/BHI.2012.6211698

[98] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, Soren Brage, Nick Wareham, and Cecilia Mascolo. 2021. SelfHAR: Improving Human Activity Recognition through Self-training with Unlabeled Data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5 (3 2021). Issue 1. DOI: http://dx.doi.org/10.1145/3448112

[99] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, and Cecilia Mascolo. 2020. Exploring Contrastive Learning in Human Activity Recognition for Healthcare. (11 2020). http://arxiv.org/abs/2011.11542

[100] Andriy Temko. 2017. Accurate Heart Rate Monitoring During Physical Exercises Using PPG. *IEEE Transactions on Biomedical Engineering* 64 (9 2017), 2016–2024. Issue 9. DOI: http://dx.doi.org/10.1109/TBME.2017.2676243

[101] Vincent Theodoor van Hees, S. Sabia, S. E. Jones, A. R. Wood, K. N. Anderson, M. Kivimäki, T. M. Frayling, A. I. Pack, M. Bucan, M. I. Trenell, Diego R. Mazzotti, P. R. Gehrman, B. A. Singh-Manoux, and M. N. Weedon. 2018. Estimating sleep parameters using an accelerometer without sleep diary. *Scientific Reports 2018 8:1* 8 (8 2018), 1–11. Issue 1. DOI: http://dx.doi.org/10.1038/s41598-018-31266-z

[102] Olivia Walch, Yitong Huang, Daniel Forger, and Cathy Goldstein. 2019. Sleep stage prediction with raw acceleration and photoplethysmography heart rate data derived from a consumer wearable device. *Sleep* 42 (12 2019). Issue 12. DOI: http://dx.doi.org/10.1093/SLEEP/ZSZ180

[103] Karl Weiss, Taghi M. Khoshgoftaar, and Ding Ding Wang. 2016. A survey of transfer learning. *Journal of Big Data* 3 (12 2016), 1–40. Issue 1. DOI: http://dx.doi.org/10.1186/S40537-016-0043-6/TABLES/6

[104] James R. Williamson, Brian Telfer, Riley Mullany, and Karl E. Friedl. 2021. Detecting Parkinson's Disease from Wrist-Worn Accelerometry in the U.K. Biobank. *Sensors 2021, Vol. 21, Page 2047* 21 (3 2021), 2047. Issue 6. DOI:http://dx.doi.org/10.3390/S21062047

[105] Hanguang Xiao, Tianqi Liu, Yisha Sun, Yulin Li, Shiyi Zhao, and Alberto Avolio. 2024. Remote photoplethysmography for heart rate measurement: A review. *Biomedical Signal Processing and Control* 88 (2 2024), 105608. DOI: http://dx.doi.org/10.1016/J.BSPC.2023.105608

[106] Jiping Xiong, Lisang Cai, Dingde Jiang, Houbing Song, and Xiaowei He. 2016. Spectral matrix decomposition-based motion artifacts removal in multi-channel PPG sensor signals. *IEEE Access* 4 (2016), 3076–3086. DOI: http://dx.doi.org/10.1109/ACCESS.2016.2580594

[107] Yang Yao, Md. Mobashir Hasan Shandhi, Jin-Oh Hahn, Omer T. Inan, Ramakrishna Mukkamala, and Lin Xu. 2022. What Filter Passband Should be Applied to the Ballistocardiogram? *SSRN Electronic Journal* (7 2022). DOI:http://dx.doi.org/10.2139/SSRN.4142412

[108] Rasoul Yousefi, Mehrdad Nourani, Sarah Ostadabbas, and Issa Panahi. 2014. A motion-tolerant adaptive algorithm for wearable photoplethysmographic biosensors. *IEEE Journal of Biomedical and Health Informatics* 18 (2014), 670–681. Issue 2. DOI: http://dx.doi.org/10.1109/JBHI.2013.2264358

[109] Hang Yuan, Shing Chan, Andrew P. Creagh, Catherine Tong, David A. Clifton, and Aiden Doherty. 2022. Self-supervised Learning for Human Activity Recognition Using 700,000 Person-days of Wearable Data. (6 2022). http://arxiv.org/abs/2206.02909

[110] Gus Q. Zhang and Weiguo Zhang. 2009. Heart rate, lifespan, and mortality risk. *Ageing Research Reviews* 8 (1 2009), 52–60. Issue 1. DOI: http://dx.doi.org/10.1016/J.ARR.2008.10.001

[111] Miao Zhang, Lishen Qiu, Yuhang Chen, Shuchen Yang, Zhiming Zhang, and Lirong Wang. 2023. A Conv -Transformer network for heart rate estimation using ballistocardiographic signals. *Biomedical Signal Processing and Control* 80 (2 2023). DOI: http://dx.doi.org/10.1016/j.bspc.2022.104302

[112] Zhilin Zhang. 2015. Photoplethysmography-Based Heart Rate Monitoring in Physical Activities via Joint Sparse Spectrum Reconstruction. *IEEE Transactions on Biomedical Engineering* 62 (8 2015), 1902–1910. Issue 8. DOI:http://dx.doi.org/10.1109/TBME.2015.2406332

[113] Zhilin Zhang, Zhouyue Pi, and Benyuan Liu. 2015. TROIKA: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. *IEEE Transactions on Biomedical Engineering* 62 (2 2015), 522–531. Issue 2. DOI: http://dx.doi.org/10.1109/TBME.2014.2359372

[114] Chao Zhao, Wenru Zeng, Dandan Hu, and Hong Liu. 2021. Robust Heart Rate Monitoring by a Single Wrist-Worn Accelerometer Based on Signal Decomposition. *IEEE Sensors Journal* 21 (7 2021), 15962–15971. Issue 14. DOI: http://dx.doi.org/10.1109/JSEN.2021.3075109

[115] Dadi Zhao, Yu Sun, Suiren Wan, and Feng Wang. 2017. SFST: A robust framework for heart rate monitoring from photoplethysmography signals during physical activities. *Biomedical Signal Processing and Control* 33 (3 2017), 316–324. DOI: http://dx.doi.org/10.1016/j.bspc.2016.12.005

[116] Johannes Zschocke, Maria Kluge, Luise Pelikan, Antonia Graf, Martin Glos, Alexander Müller, Rafael Mikolajczyk, Ronny P. Bartsch, Thomas Penzel, and Jan W. Kantelhardt. 2019. Detection and analysis of pulse waves during sleep via wrist-worn actigraphy. *PLOS ONE* 14 (12 2019), e0226843. Issue 12. DOI: http://dx.doi.org/10.1371/JOURNAL.PONE.0226843