

Validation of a Wrist-Worn Photoplethysmographic Sensor (E4): Comparison of Finger and Wrist

N.M. Enewoldsen | s1481436

Enschede, September 2021

Master Thesis

Human Factors & Engineering Psychology

Faculty of Behavioural, Management & Social Sciences

1st Supervisor: Prof. Dr. J.M.C. Schraagen

2nd Supervisor: Dr. R.H.J. van der Lubbe

UNIVERSITY OF TWENTE.

Table of Contents

Abstract	4
List of Abbreviations.....	5
Introduction	6
Methods	11
Participants.....	11
Materials	12
Design	14
Procedure	15
Data Analysis	15
Results	17
STAI Questionnaire Analysis	17
Signal Comparison.....	19
Parameter Comparison.....	20
Event Comparison.....	21
Discussion	24
Subjective Stress Measurement	25
Signal Level	26
Parameter Level	28
Event Level	28
Limitations	30
Conclusion	31
References	32
Appendix	43
Appendix A – Screening Questions.....	43
Appendix B – Experiment Script (Psychopy).....	44
Appendix C – Normalized STAI-Scores Graphic (R-Script)	64

VALIDATION OF A WRIST-WORN PPG SENSOR

Appendix D – Visual Inspection Graphs (R-Script)	65
Appendix E – Pre-processing & Extraction of Parameters (Python-Script).....	70
Appendix F – Bland-Altman Plot (R-Script)	97
Appendix G – Error Bar Plot (R-Script)	98
Appendix H – Overview of mean STAI scores of participants per phase	99

Abstract

Background: The fast rise of wearable technology has led to the development of photoplethysmography (PPG) sensors which enable the monitoring of physical and emotional states (e.g., stress) of users in real-time and in daily life. Thereby, detection and management of stress is a key component of the research, due to the detrimental negative effects of stress on health. However, studies have shown the sensors' vulnerability towards artifacts, especially motion artifacts, which emphasize the need for accurate validity assessments. The present study implements the validity assessment protocol of van Lier et al. (2019), whereby the wrist-worn PPG sensor of the Empatica E4 is validated by comparing it to a PPG sensor at the fingertip which reflects the reference device (RD).

Methods: A modified version of the Sing-A-Song-Stress task (SSST) and the Stroop task were used as social and cognitive stressor, in order to induce different intensities of stress. Beside the recording of the physiological arousal, the State Trait Anxiety Inventory (STAI) was used to measure the subjective stress experience. The analysis was performed at the signal, parameter and event level, enabling a comprehensive and standardized validation of the E4.

Results: The successful induction of subjective stress for both tasks could not be measured by the physiological measurements. On the one hand, the strict quality inclusion criteria for the analysis resulted in the exclusion of around 40% of the data, while on the other hand, the stressors were found to be not strong enough to induce detectable physiological arousal. Furthermore, at the signal level of the PPG, no relationship was found between the measurements of the two sensors, which is supported by the differences in signal, demonstrated by the analysis at the parameter level.

Conclusions: This study questions the validity of the E4 by showing that the validity is lower than described in the literature. Nevertheless, previous research has shown that the E4 can be used for strong and long-lasting stressors, as well as for averages of HR over a longer period of time. Moreover, the validity assessment protocol showed its potential, by enabling the less time-consuming validation with a PPG sensor as RD instead of an electrocardiogram, consequently, making the protocol more available and less restricting with regard to participants and setting.

Keywords: Wearable Sensor, Photoplethysmography, Heart Rate, Stress, Validity

VALIDATION OF A WRIST-WORN PPG SENSOR

List of Abbreviations

ANS	Autonomous Nervous System
BP	Blood Pressure
BS1	Baseline (first)
BS2	Baseline (second)
BVP	Blood Volume Pulse
CVS	Cardiovascular System
ECG	Electrocardiogram
HR	Heart Rate
HRV	Heart Rate Variability
PNS	Parasympathetic Nervous System
PPG	Photoplethysmography
PPI	Peak-to-Peak Interval
RC1	Recovery Phase (first)
RC2	Recovery Phase (second)
RD	Reference Device
RMSSD	Root Mean Square of Successive Differences
SDNN	Standard Deviation of Peak-to-Peak Interval
SNS	Sympathetic Nervous System
SQI	Signal Quality Index
SSST	Sing-a-Song Stress Task
ST1	Stress Task (first; modified. SSST)
ST2	Stress Task (second; Stroop Task)
STAI	State Trait Anxiety Inventory
UT	University of Twente

Introduction

Wearable sensors are becoming more widely available and cheaper, offering new opportunities in healthcare by enabling the monitoring of the physical, as well as physiological changes related with the emotional state in daily life and in real-time (Bonato, 2003; Elgendi, 2012; Fletcher et al., 2011; Picard & Healey, 1997; Ragot et al., 2017). These devices are available in many different shapes and forms, equipped with different combinations of sensors (Gu et al., 2009; Poh et al., 2010; Poon et al., 2006). By measuring these physiological parameters in daily life, health informatics can use this information to detect, prevent or treat diseases (Cook et al., 2006; Wilard et al., 2011; Zheng et al., 2014).

Thereby, the measurement of mental stress has become a key component which is associated with physiological and psychological reactions (Chrousos, 2009). Specifically, “stress can be conceptualized as the response of the Autonomic Nervous System (ANS) to various stimuli” (Nath et al., 2020, p.1) including physical, psychological and environmental stimuli (Selvaraj, 2015). There are two types of stress responses of the ANS, namely, the physiological response, which is the effect of stress by physiological activities (e.g., increased heart rate, increased cortisol level, increased sweating), as well as the psychological response which reflects the stress perceived by the individual (e.g., mental stress, emotional stress, perceptual stress) (Nath et al., 2020). In the short term, the activation of stress responses is adaptive, but can become maladaptive if stress responses are repeatedly provoked (Kemeny, 2003). These maladaptive physiological and psychological stress responses can result in various disorders (e.g., hypertension, cardiovascular diseases, anxiety disorders) (Chrousos, 2009; Yoo & Lee, 2011; Zangróniz et al., 2018). Therefore, detection and management of mental stress is essential, in order to prevent those negative effects of mental stress and reduce health risks (Mitsuhashi et al., 2019; Nath et al., 2020; Yoo & Lee, 2011). For the effective detection of stress, various vital signs which were shown to be sensitive to changes in mental stress, are measured by wearables (Abbod et al., 2011; Mitsuhashi et al., 2019; Singh et al., 2011; Yoo & Lee, 2011; Zangróniz et al., 2018).

The pursuit of practical and valid methods to measure stress is reflected by the adoption of multi-sensor technology and the integration of reflective photoplethysmography (PPG) sensors which describe the “optical detection of pulsatile blood volume changes in the vascular bed under the PPG sensor” (Greve et al., 2012, p.1; Elgendi, 2012). Specifically, the PPG sensor measures changes in tissue and blood volume by emitting light into tissue and detect optical variations, in form of light reflected from or transmitted through the tissue (Fusco et al., 2015; Santos et al., 2012; Zhang et al., 2019; Zheng et al., 2014). Compared to conventional

VALIDATION OF A WRIST-WORN PPG SENSOR

plethysmograph methods, the reflective PPG is a non-invasive method, enabling a more practical and user-friendly experience, as well as the use of those sensors in an ambulatory setting.

Although studies have investigated the PPG signal and demonstrated satisfactory data quality, high quality could only be achieved in the absence of excessive hand movements or during resting phases (McCarthy et al., 2016; Pietilä et al., 2017). Specifically, the quality of the recorded data is susceptible to contamination, so-called artifacts, which have to be removed, in order to obtain a high quality. With regard to the PPG sensor, their sensitivity to the correct placement of the sensor, movements (physical exercise) and cardiac arrhythmia can influence the rate parameter and reduce its reliability if these artifacts are not removed properly (Allen, 2007; Mullan et al., 2015; Picard & Healey, 1997; van Dooren et al., 2012; Wander & Morrison, 2014; Zheng et al., 2014). Even “the properties of the subject’s skin at measurement, including the individual skin structure, the blood oxygen saturation, blood flow rate, skin temperature and the measuring environment“ (Elgendi, 2012, p.16) can have a negative impact on the PPG signal. Furthermore, measurements of the PPG sensor are influenced by factors such as emotional states, ambient temperature and fitness level, and they respond and re-stabilize only slowly to them (Picard & Healey, 1997; Tapia et al., 2007). Therefore, several algorithms have been developed which try to remove noise as accurately as possible, whereby some even integrate additional recorded acceleration data as an indicator of the actual movements of the user (Mullan et al., 2015).

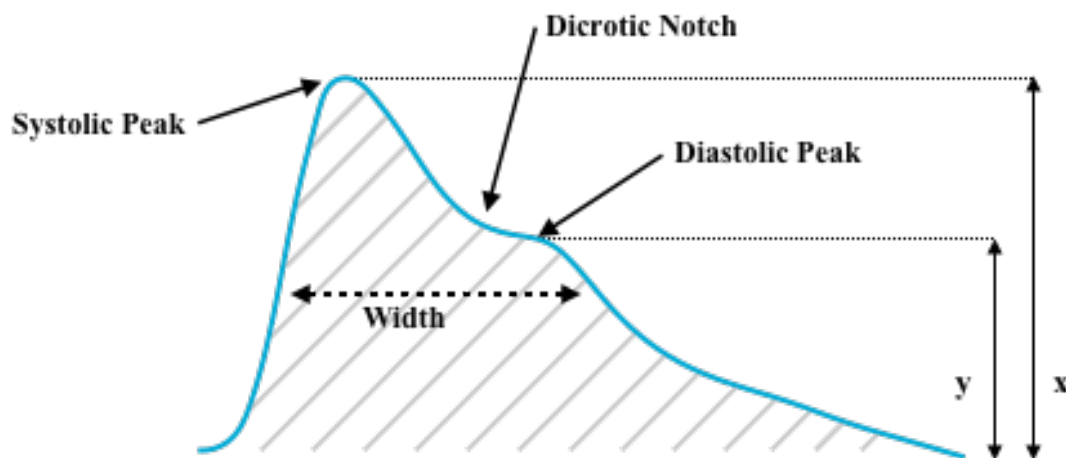
Summarized, the susceptibility of wearable PPG sensors towards artifacts and the rapid development of new alternatives emphasize the need for systematic validation of those sensors (Goldsack et al., 2020; Jo et al., 2016; Munos et al., 2016; Shcherbina et al., 2017). Therefore, the measurement of the PPG sensor of interest is compared with the “true” measurement of an already established measurement device, also known as reference device (RD). In particular, the detection of physiological changes due to external stressor levels, measured by the RD and the device of interest, enable a comprehensive validation on multiple levels. The physiological changes, also called events, can be examined by evaluating and comparing the ability of both sensors to significantly detect those specific events (van Lier et al., 2019). Furthermore, the parameter level focusses on whether the similar physiological parameters can be extracted from the recordings of the new device, compared to the recordings of the RD. Finally, the signal level reflects the extent to which the device of interest and the RD produce similar signals which can be determined by assessing the recorded raw data.

VALIDATION OF A WRIST-WORN PPG SENSOR

Among the measured signals of the PPG is the heart rate variability (HRV), a “popular non-invasive marker of the autonomic nervous system and is widely used to assess cardiac health” (Jeyhani et al., 2015, p.1; Elgendi et al., 2010). HRV reflects the variance of time between consecutive heartbeats (i.e., beat-to-beat variation, also known as inter-beat interval or peak to peak interval; PPI) and provides an indirect insight into the cardiovascular system (CVS) which controls the blood flow through the body, and which can be characterized by measurable parameters such as heart rate (HR), peak to peak interval (PPI), blood pressure (BP) and blood volume pulse (BVP) (Akhter et al., 2016; Mandryk et al., 2006). These cardiovascular variables are fluctuating on a beat-to-beat basis and are coordinated by balanced activity of the two branches of the ANS, the parasympathetic nervous system (PNS) which is responsible for the restoration and conservation of bodily energy, and the sympathetic nervous system (SNS) which increases metabolic output to deal with external challenges and mobilizes the body system during activities, as well as stressful situations (Akhter et al., 2016; Appel et al., 1989; Poh et al., 2010). Thus, in preparation for motor action, the sympathetic arousal elevates the HR, BP, sweating and redirects blood toward skeletal muscles, lungs, the heart and the brain, which are also activated in stressful situations when the body is in a fight or flight state (acute stress response) (Yoo & Lee, 2011). By measuring BVP which corresponds to changes in blood flow and HR, the PPG sensor enables the measurement of cardiovascular changes in the ANS caused by stress, as well as the computation of HRV parameter like PPI and HR. This is due to the pulsatile component of the PPG signal which is synchronous with

Figure 1

A typical PPG waveform with its characteristic parameters



Note. The x reflects the amplitude of the systolic peak and y the amplitude of the diastolic peak.

VALIDATION OF A WRIST-WORN PPG SENSOR

the fluctuations of the beating heart and thus visible in the rise and fall of the PPG signal (Allen, 2007).

The waveform of the PPG signal can be divided into two phases (see Figure 1), the anacrotic phase which is primarily concerned with systole reflection, and the catacrotic phase which describes the diastole and wave reflections (Elgendi, 2012). Beside the parameters shown in Figure 1 (e.g., the systolic amplitude and the pulse width), important features of the PPG signal include the pulse area which reflects the total area under the PPG curve (indicated by the grey lines in Figure 1) and the peak to peak interval (PPI).

With regard to validity assessment of PPG sensors, most research is conducted by comparing the electrocardiogram (ECG), also known as the golden standard of HRV, with the PPG recordings. Whereas RR intervals are a prominent feature of the ECG signal which describes the distance between two successive R peaks, the PPG signal has PPIs which represent the time between the top of the peaks in the BVP signal (Akhter et al., 2016). A great number of publications focused on the correlation between RRs and PPIs, as well as the HRV parameters derived from PPG and ECG. The majority of these publications demonstrated a high correlation and good agreement with regard to PPG and ECG parameters in various conditions (Bolanos et al., 2006; Elgendi, 2012; Giardino et al., 2002; Greve et al., 2012; Jeyhani et al., 2015; Lin et al., 2014; Selvaraj et al., 2008). Gil et al. (2010) even suggested “the use of the PRV signal as an alternative measurement of the HRV signal during non-stationary conditions” (p.1288). However, the high correlation between PPG and ECG seems to decrease in conditions where participants have to exercise which is consistent with the sensitivity issue of PPG sensors (Lin et al., 2014). Only a few studies were unable to establish a statistical correlation (Jo et al., 2016; Parasnis et al., 2015).

Although these studies all examined the validity of PPG sensors, they often come to inconclusive and incomparable inferences, due to the use of different statistical methods (Sartor et al., 2018; Zaki et al., 2012), evaluation on different variable levels (Shcherbina et al., 2017) and the lack of decision criteria to determine the validity of those sensors (van Lier et al., 2019), indicating a lack of standardized validity assessments. Therefore, van Lier et al. (2019) introduced a standardized validity assessment protocol for physiological signals (e.g., electrodermal activity and cardiovascular activity) on three levels: the signal, parameter and event level. The signal level assesses the ability of a device to measure similar signals, compared with a reference device (RD), and whether these differences lie within an acceptable range with regard to their correlations (van Lier et al., 2019). The parameter level focusses on whether the device is able to produce the similar physiological parameters for each individual,

VALIDATION OF A WRIST-WORN PPG SENSOR

compared to the RD, while the event level determines the ability to detect events reflected by physiological changes, due to external stressors.

In particular, van Lier et al. (2019) validated the PPG sensor of the E4 wearable by comparing it with an ECG. In contrast to extracted parameters of the PPG and ECG signal which can be correlated, the signal themselves have quite different characteristics, due to the different measurement techniques. Therefore, the comparison at the signal level was not performed since it would be impossible to determine which differences were introduced by the wearable and which was from using different techniques. In order to implement the validity assessment protocol and evaluate a PPG sensor at all three levels, a validation of such a PPG sensor has to be compared with a PPG sensor which is already validated and established. Subsequently, the PPG-based measurement of CVA at the fingertip is a conventional stress measurement approach in ambulatory settings and, therefore, a practical RD-alternative to the ECG (Nardelli et al., 2020). Due to the fact that the validity assessment protocol was only applied with an ECG as RD, the complete protocol has to be implemented with a PPG as RD, in order to be able to make a reliable statement about the validity of the assessed sensor. Moreover, this would not only demonstrate the ability of the assessment protocol to be used with RDs other than the ECG, but also enable a more efficient and cheaper way of validating sensors, due to the availability and easy usage of PPG sensors. The accessibility of PPG sensors in comparison with the ECG, combined with the standardized validity assessment protocol, could facilitate a more frequent and straightforward approach to validating sensors.

In this article, the validity of the wrist-worn E4 PPG sensor is assessed by applying the comprehensive validity assessment protocol of van Lier et al. (2019) and by comparing the E4 to an optical pulse sensor which measures PPG from the fingertip. Therefore, a thorough investigation at the signal, parameter and event level is performed. By inducing different intensities of stress, in the form of a social and a mental task, the sensors can be observed in various situations, enabling a comprehensive analysis.

Specifically, the social and mental stress tasks were found to be the most effective stressors, whereby the immediate affection of the psychophysiological responses by these stressors is guaranteed (Dickerson & Kemeny, 2004; Linden et al., 1997). On the one hand, a modified version of the Sing-a-Song Stress Test (SSST) is used to induce social stress in an ethical and straightforward approach (Brouwer & Hogervorst, 2014). The magnitude of increased stress levels by the original SSST are comparable to the Trier Social Stress Test (Kirschbaum et al., 1993). However, the SSST can be performed by one researcher in a shorter amount of time than the Trier Social Stress Test, which is an effective, but time-consuming

VALIDATION OF A WRIST-WORN PPG SENSOR

task. Therefore, the SSST is chosen over the Trier Social Stress Test, due to its quick and feasible approach (van Lier et al., 2019). Moreover, Egilmez et al. (2017) investigated several stress-inducing activities, like a Math test and the Stroop Color-Word Test, and showed that singing induced the most stress in young adults. Nonetheless, studies investigating the effects of the SSST on HR indicated that the main increase in stress was observed during the preparation phase, before participants had to actually sing (Kudielka et al., 2004; von Dawans et al., 2011). In particular, Kudielka et al. (2004) demonstrated that the HR response pattern increased significantly during the preparation phase for children, younger adults and older adults. The significant increase in stress during the preparation phase could also be shown in subjective stress measurements (Häusser et al., 2012; von Dawans et al., 2011). Therefore, the SSST is modified, in order to be even more time efficient, and measure a fast increase induced by social stress, as well as a relatively abrupt decrease in form of relief. On the other hand, the Stroop color-word task is chosen as a demanding cognitive stressor (Choi et al., 2010; Egilmez et al., 2017; Manuck et al., 1996; Muldoon et al., 1992; Poh et al., 2010). Although van Lier et al. (2019) used a noise task, they recommended to use a stronger and longer stressor, due to the fact that they could not distinguish the responses measured by both sensors. In particular, the Stroop task is likely to increase the stress slowly, due to its permanent cognitive demand, followed by a slow decline, caused by its repetitiveness (Al'Absi et al., 1997; Kamarck & Lovallo, 2003). It was chosen to do the cognitive task last, due to its potential to provoke anger or frustration in participants which was shown to prolong the return to the baseline levels, whereby even 10 minutes of recovery were not enough to return to baseline levels (Baum & Contrada, 2010; Linden et al., 1997).

Methods

Participants

Twenty healthy participants (age: $M = 22.9$, $SD = 2.49$; gender: 65% female) participated in the experiment. The study protocol was approved by the Ethics Committee of the University of Twente (UT) and written consent was obtained from the participants beforehand. Recruitment was based on convenience sampling, involving approaching people on the campus of the UT and using the university internal SONA platform which provides undergraduate students with a full overview of the study, as well as the opportunity to sign up and choose their preferred timeslot. Thereby, participants had to be able to physically perform the exercises with minimal risk of health complications. In case of reported significant medical conditions, especially diseases which are related to the cardiovascular system, participants were excluded (see Appendix A). The exclusion criteria also included participants dealing with

VALIDATION OF A WRIST-WORN PPG SENSOR

addictions or the consumption of drugs, medication or substances which have a direct or indirect impact on the CVS including alcohol (Quintana & Heathers, 2014). Although smoking cigarettes and drinking coffee was not an exclusion criterion, participants were asked beforehand to avoid consuming a few hours before the experiment. These criteria enabled a high reliability by minimizing factors which might have disrupted or distorted the recordings.

Materials

The experiment was programmed with Python 2.7 and PsychoPy 3 (Peirce et al., 2019) (see Appendix B). The interaction between participant and program was restricted to key input. This applied to the questionnaires, the tasks, as well as the instructions which were automatically presented on a 15" Windows 7 laptop. In addition, the program saved timestamps for every interaction, as well as starting points of important parts to separate files.

Empatica E4

The Empatica E4 (Empatica, Milan, Italy) is a wearable device designed for continuous, real-time data acquisition in daily life. Beside the PPG sensor, the device is able to measure several different psychophysiological responses of the body, like electrodermal activity, temperature and acceleration. The PPG sensor measures BVP with a sample rate of 64 Hz which is used in order to compute the HR and PPI. Furthermore, the sensor is equipped with an internal real-time clock, enabling recordings for over 36 hours with a capacity of over 60 hours of data storage. Specifically, the data is stored on the internal memory of the E4 and can be downloaded via USB through the software Empatica Manager. Empatica Connect, a web-application, is then needed to download the raw data in CSV-format. With regard to the assessment of cardiac activity, a validation study of the E4 demonstrated sufficiently precise PPG signals (McCarthy et al., 2016). Moreover, Ragot et al. (2017) concluded that the wearable devices, and especially the E4, “appear as accurate as laboratory sensors for emotion recognition” (p.20).

Shimmer3 GSR+

The Shimmer3 GSR+ unit (Shimmer Sensing, Dublin, Ireland) is a wireless biosensor designed for simultaneous and real-time measurements. In particular, the Shimmer3 is able to capture electrodermal activity and acceleration, as well as continuous heart rate via an optical pulse sensor. Therefore, the optical pulse sensor has to be attached to the GSR+ module via the 3.5 mm headphone port, providing a PPG signal from the finger with a sample rate of 128 Hz. The recorded data is stored locally on microSD and has to be imported with the Consensys Software which also allows for the transformation of PPG data into HR. In order to transfer the recordings to a computer, the sensor has to be connected with the docking station, which is also used for charging, as well as calibrating the sensor.

Short State-Trait Anxiety Inventory

The short version of the State-Trait Anxiety Inventory (STAI; Spielberger et al., 1983) consists of six items on a four-point Likert scale (see Table 1). Specifically, this questionnaire was used to determine the perceived stress of the participants before and after each task, in order to investigate the relationship between the subjective levels of stress and the recordings of the sensors. The original STAI is used to measure „the presence and severity of current symptoms of anxiety and a generalized propensity to be anxious“ (Julian, 2011, p.467), whereby one of the leading interests in the research domain is the investigation of its possible ability to indicate the presence of mental disorders (Groth-Marnat, 2003; Julian, 2011; Kvaal et al., 2005). Consequently, the short STAI consists of three anxiety-absent items and three anxiety-present items which were identified by Spielberger, one of the main developers of the original STAI, to be particularly sensitive to low and high stressors, respectively (Marteau & Bekker, 1992; Spielberger et al., 1983). According to Marteau and Bekker (1992) who developed and validated the short version of the STAI, this shortened questionnaire has a good reliability (reliability coefficient of 0.82) and validity which was replicated by Tluczek et al. (2009).

Table 1

Self-evaluation questionnaire (short STAI: Y-6 item) (Marteau & Bekker, 1992)

Items	Not at all	Somewhat	Moderately	Very much
1. I feel calm ...	1	2	3	4
2. I am tense ...	1	2	3	4
3. I feel upset ...	1	2	3	4
4. I am relaxed ...	1	2	3	4
5. I feel content ...	1	2	3	4
6. I am worried ...	1	2	3	4

Modified Sing-a-Song Stress Test

A modified version of the Sing-a-Song Stress Test (SSST) was used to induce social stress in an ethical and straightforward approach (Brouwer & Hogervorst, 2014). Due to studies

VALIDATION OF A WRIST-WORN PPG SENSOR

demonstrating a significant increase in stress during the preparation phase, the modification included that participants only were shown the message to prepare to sing a song, but not actually had to sing a song (Kudielka et al., 2004; von Dawans et al., 2011). This was implemented, in order to measure a steep decline in stress through the relief of not having to sing a song which would be more gradually if participants should have to sing a song. This way, a steep increase, as well as steep decrease in stress could be measured. Participants were asked not to disclose the fact that they do not actually have to perform the song to other participants, in order to prevent someone from knowing this beforehand which would make the social stress induction ineffective.

During the task, participants got the instruction to think of a song which they later had to perform for 30 seconds. They had two minutes to think of a song which could have been any song they liked or knew. After the anticipation period, a message was displayed which requested the participants to prepare for singing (“Are you ready?”) and press space to continue. Instead of the instruction to start singing, participants were informed that they do not have to sing, in order to be able to measure the relief in the form of a fast decrease of cardiovascular activity.

Stroop Color-Word Test

The Stroop Color-Word Interference Test is a demanding task designed to induce psychological stress and physiological arousal (Choi et al., 2010; Egilmez et al., 2017; Manuck et al., 1996; Muldoon et al., 1992; Poh et al., 2010). For each trial, one of three color words (red, green or blue) was displayed in one of those colors (red, green or blue), whereby both, word and color, were randomly selected. At the bottom of the screen, the three words were presented in their matched color and with their related key on the keyboard (Red: B, Green: N, Blue: M). Participants had to press the key that matched the color of the presented word. Furthermore, instructions were provided with a short test run which displayed feedback in the form of a green window for the correct key and a red window for pressing the wrong key. In total, participants had to execute 500 trials which took approximately eight minutes. The responses of those trials were neither registered, nor analyzed.

Design

For the present study, a within-subjects design was implemented, whereby all participants were equipped with both, the Shimmer3 and the Empatica E4, sensors. Thereby, all participants had to perform the modified SSST and the Stroop task, in order to induce different levels of stress (Dickerson & Kemeny, 2004; Linden et al., 1997).

Procedure

In the beginning of each session, participants were brought to a quiet room and sat in front of the laptop to acclimate while they were briefed by the Experimenter regarding the experiment. They were informed on how their data would be used, that participation was voluntary and that they could stop at any point during the experiment. It was emphasized that participants should move as little as possible, especially with the arm where both sensors would be attached to. Also, they were informed that the experiment would take approximately 40 minutes and that instructions were displayed on the screen. Afterwards, they were asked to read and sign the informed consent.

Both sensors were then attached to the non-dominant arm of the participant, so that they still could press keys with their dominant hand without producing a wide range of artifacts. In particular, the E4 was placed on the wrist, while the Shimmer3 was clipped at the index finger. Next, the program with the experiment was started and the participants demographics were noted. There was a baseline (BS1) and recovery phase (RC1) of 5 minutes before and after the modified SSST (ST1). With regard to the Stroop task, there was also a second baseline (BS2) before the Stroop task (ST2), followed by a recovery phase (RC2). Participants were instructed to sit quietly, move as little as possible and focus on their breathing. In addition, participants filled out the short STAI questionnaire before and after each task, as well as after both recovery periods. In order to increase the stress during the modified SSST, the researcher took a seat next to the participant. For the Stroop Word-Color Test, as well as the rest of the experiment, the researcher was in the room next to the participant. After completing the experiment, the researcher removed both sensors and thanked participants for their participation.

Data Analysis

STAI Questionnaire Analysis

STAI scores were calculated for each of the six-time measurements in 20 participants, as acute stress reference measurement. The statistical significance of the STAI scores between pre- and post-SSST, as well as pre- and post-Stroop task was determined by using a paired sample t-test. The STAI scores for each measurement point were normally distributed, as assessed by the Shapiro-Wilk Test ($W_{T1 \text{ before SSST}} = .929$, $p_{T1 \text{ before SSST}} = .150$; $W_{T2 \text{ after SSST}} = .952$, $p_{T2 \text{ after SSST}} = .394$; $W_{T3 \text{ after Recovery}} = .906$, $p_{T3 \text{ after Recovery}} = .054$; $W_{T4 \text{ before Stroop}} = .919$, $p_{T4 \text{ before Stroop}} = .095$; $W_{T5 \text{ after Stroop}} = .943$, $p_{T5 \text{ after Stroop}} = .269$; $W_{T6 \text{ after Recovery}} = .936$, $p_{T6 \text{ after Recovery}} = .205$). The graph was created with the statistical program R (R Core Team, 2020) and the associated script can be found in appendix C.

Data Quality Assessment

VALIDATION OF A WRIST-WORN PPG SENSOR

The recordings were visualized for each participant and phase after the completion of the experiment with the statistical program R (R Core Team, 2020), in order to get an overview with regard to the quality of the data (see Appendix D). Due to artifacts and the fluctuating stability of the PPG signal introduced most likely by motions and the sensitivity of the sensors itself, there were parts of participants' data which had to be removed because they contained too many artifacts to be interpretable (Elgendi, 2012). However, the quality had to be assessed carefully, in order to only remove parts with a low signal-to-noise ratio, while taking into account the limited amount of recorded data. Therefore, a signal quality measure in form of a signal quality index (SQI; Karlen et al., 2012; Orphanidou et al., 2014) was applied throughout the whole PPG signals and for each of the important phases (baselines, recovery periods, stress tests), ranging from 0 to 100 (van Lier et al., 2019). For the baselines, as well as the two main tasks, a signal quality of 70 was needed in order to be included in the further analyses. If a certain recording of one of the phases had less than 50% of usable data after the removal of artifacts, it was also excluded because there was not enough reliable data to make a reliable interpretation. In addition to the signal quality measure, the PPG signals should be checked for powerline interference artifacts which could be due „to the instrumentation amplifiers, the recording system picking up ambient electromagnetic signals and other artifacts“ (Elgendi, 2012, p.17), high frequency artifacts, arrhythmia, motion as well as muscle artifacts and low amplitudes. In advance of the application of the signal quality index, the data of three participants were removed, due to unrealistic values measured by the E4. In particular, for each of the three participants the PPG signal dropped around zero for the majority of the recordings which made it not usable for the further analysis. The signal quality of the remaining 17 participants was on average 82.7 ($SD = 10.9$). This approach led to the removal of 37% of the data, whereby the complete datasets of further three participants were removed. Thereby, the majority of signal quality indices below 70 were found in measurements of the E4. In total, the data of 14 participants were used for further analysis.

Signal Comparison (Cross-correlation Function)

In order to perform a cross-correlation function on both PPG signals, the recordings of the Shimmer3 had to be sampled down from 128 Hz to 62.5 Hz. Therefore, a python script was written for the downsampling, as well as the computation of the cross-correlation coefficients across different time lags (see Appendix E). For each of the important phases, cross-correlation was performed, whereby correlations were considered very strong (>0.9), strong ($0.7-0.9$), moderate ($0.5-0.7$) or weak (<0.5), depending on the coefficient. However, only correlations of 0.8 or higher are considered acceptable (van Lier et al., 2019).

Parameter Comparison (Bland Altman Plot)

After downsampling, both PPG signals were filtered with low-pass and high-pass filters between 5 and 15 Hz. The PPIs were produced by calculating the duration between successive peak locations, whereby only PPIs with a length between 0.33 and 1.5 were included (van Lier et al., 2019). Next, the signal quality index (Orphanidou et al., 2014) was assigned to the PPIs, only when at least 50% of the data had an SQI of 80%. In order to compare both sensors on a parameter level, three commonly retrieved parameters of the time domain were calculated, namely the mean PPI which was converted into instantaneous HR, the SD PPI (SDNN) and the root mean square of the successive differences (RMSSD). Furthermore, the data was checked for normality and missing data. Lastly, the Bland-Altman plots were created with the “BlandAltmanLeh” package (Lehnert, 2015) for the statistical program R (R Core Team, 2020) (see Appendix F). Thereby, limits of agreement for each parameter were not to exceed 10% of the plausible values (CI) for the parameters, in order to suggest that the parameter was validly measured. Additionally, the a priori defined boundaries for HR were ± 5 bpm, the SDNN ± 0.06 and RMSSD ± 0.07 , respectively (van Lier et al., 2019).

Event Comparison (Error Bar Plot)

After downsampling the Shimmer3 data and filtering the recorded PPG signals, PPIs and HR were calculated. Furthermore, the data was checked for normality and missing data, before creating the event difference plots with R (R Core Team, 2020) (see Appendix G). In particular, a line plot with the mean and the standard error (SE) per task was created, whereby each individual is represented by a line, as well as a line plot with the differences between both sensors for each person with the mean and the SE per task. Thereby, it was important that within the same PPG sensor, the error bars should not overlap between baseline and task. In addition, the a priori defined boundaries which are the reference effects (difference between the baseline and stress task of the S3) of the HR were calculated and plotted. Specifically, the reference effects for the modified SSST and the Stroop task were ± 6.809 and ± 13.33 , respectively.

Results

STAI Questionnaire Analysis

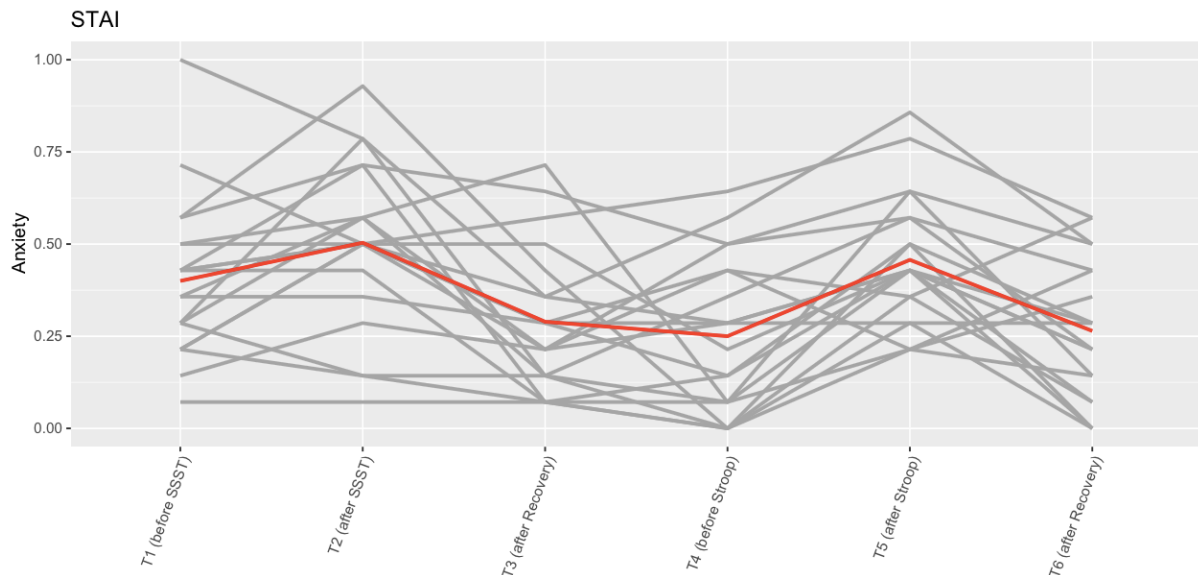
A detailed overview of the total STAI scores per phase for each participant can be found in the appendix H. The paired sample t-test found a significant difference of the subjective stress measurement before and after the modified SSST ($t(19) = -2.398, p = .027$). The inspection of the means showed an increase in stress after the task ($M = 43.50, SD = 10.51$), compared to the measured stress before the modified SSST ($M = 38.66, SD = 9.94$). With regard to the Stroop task, there was also a significant difference present in the subjective stress measurement before

VALIDATION OF A WRIST-WORN PPG SENSOR

and after the task ($t(19) = -5.040, p < .001$). In particular, an increase in stress was visible after the Stroop task ($M = 41.33, SD = 8.47$) compared to before the task ($M = 31.66, SD = 9.76$).

Figure 2

Normalized subjective measurements (total STAI scores)



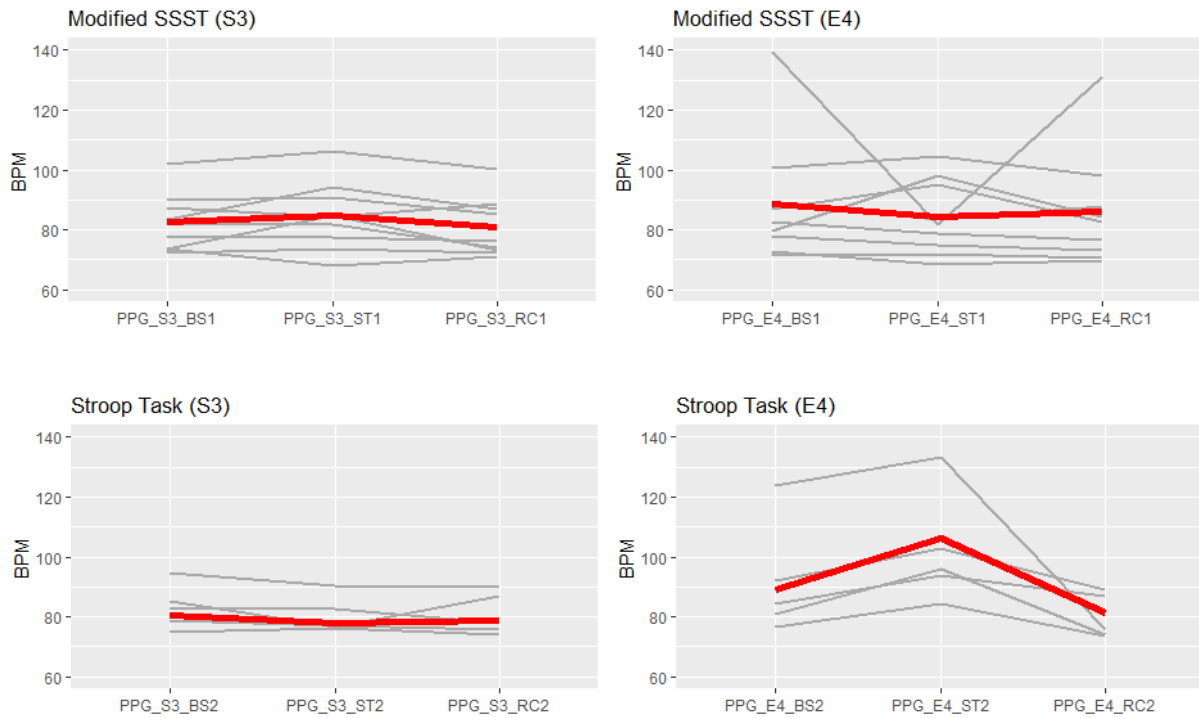
Note. The normalized subjective measurements (total STAI scores) are displayed for the relevant phases for each participant (grey lines) with the average (red line).

These results are clearly visible by the visualization of the subjective stress (see Figure 2). Thereby, a clear increase in the subjective stress can be seen in the measurement after the modified SSST, followed by a decrease during recovery and even lower stress levels after the second baseline. With regard to the Stroop task, the induction of stress had a more significant effect which is represented by a steeper increase in the average of subjective stress levels.

However, the inspection of the physiological data showed that the increase in stress could not be demonstrated by both sensors. In particular, the E4 was not able to measure the increase for the modified SSST, in fact the average showed a slight decrease in HR for ST1 (see Figure 3). In contrast, the S3 was able to measure a slight increase in HR for ST1. With regard to the Stroop task, the E4 measured on average an increase from more than 15 bpm for ST2 (see Figure 3). However, the S3 was not able to detect an increase and actually displayed a slight decrease in HR for ST2.

Figure 3

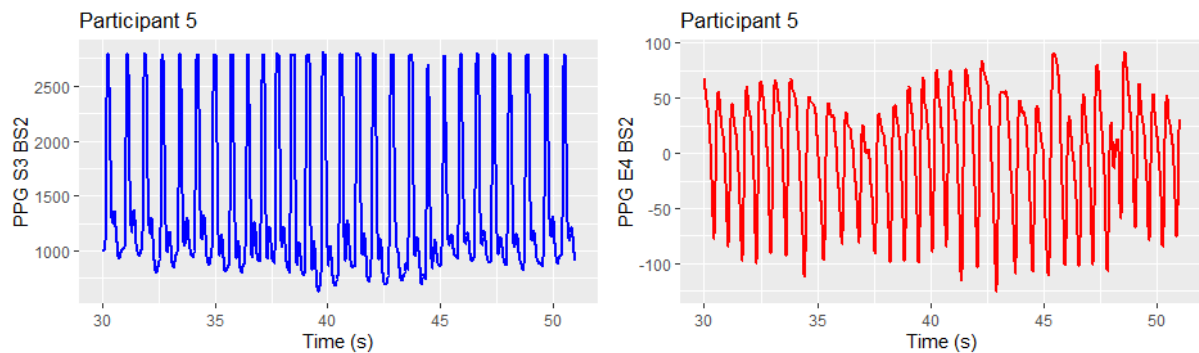
HR of the E4 and S3: modified SSST & Stroop task



Note. The measured HR of the E4 and S3 is displayed for each participant (grey lines), as well as the average (red line), with regard to the relevant phases of the modified SSST and Stroop task, respectively.

Figure 4

PPG signals of S3 and E4 of participant 5 during the second baseline



Signal Comparison

For all participants, low correlations for the measurements of the S3 and E4 were found, ranging from -0.086 to 0.097 with an average around zero ($M = 0.004$). The correlation coefficients of each participant for the PPG data of the E4 and the S3 can be seen in table 2.

VALIDATION OF A WRIST-WORN PPG SENSOR

The weak correlations can be illustrated in an example of the recorded PPG data curves of both sensors (see Figure 4). In the figure, the difference between both sensors can be seen clearly with the S3 displaying a much higher amplitude of the systolic peak, in comparison to the measured amplitude of the E4.

Table 2

Correlation Coefficients of E4 and S3 PPG data per Participant and phase

Participant	BS1	ST1	RC1	BS2	ST2	RC2
P1	-0.015	0.052	0.027	-0.006		-0.004
P2	0.042	0.025	-0.004	0.010	0.015	-0.023
P4	0.043	0.097	-0.003	0.070		-0.051
P5	0.021	0.041	-0.035	-0.037	-0.002	-0.038
P6	-0.021	-0.086	0.044	-0.042	0.024	
P7	0.035	-0.010	0.024	-0.030	-0.016	-0.044
P8		0.007				
P9	-0.042	-0.013	0.059	-0.021	0.066	
P12	-0.013	0.010		0.044		
P13				0.006		-0.058
P14	0.019		0.014	0.005	-0.023	
P15	0.063	0.091	-0.004	-0.033	-0.022	-0.022
P17	0.049		-0.005			0.046
P18	-0.030	-0.012	0.018	0.000	-0.045	0.022

Note. The correlation coefficients for E4 and S3 are displayed for the relevant phases (BS1 = first baseline; ST1 = modified SSST; RC1 = first recovery phase; BS2 = second baseline; ST2 = Stroop task; RC2 = second recovery).

Parameter Comparison

The Bland-Altman plots of the mean, SD, and the RMSSD of the PPI are displayed in Figure 5. For the HR, the majority of values are within the limits of agreement, with one participant exceeding these limits, while still being within the 95% confidence level of the limits of agreement. More than half of the participants are also within the proposed boundaries of ± 5 bpm. Thereby, all values outside the boundaries are greater than zero, showing that for

VALIDATION OF A WRIST-WORN PPG SENSOR

higher averages of HR for both sensors, the E4 measures much greater values for HR, resulting in a greater difference between the measured HR of E4 and S3. With regard to the SDNN and RMSSD of the PPIs, no values lie within the ± 0.07 boundary and ± 0.06 boundary, respectively. However, all the values of the RMSSD are within the limits of agreement, with lower means being more frequent in the 95% confidence interval of the mean difference. With regard to the SDNN Bland-Altman plot, all participants, except for one, lie within the limits of agreement, whereby the outlying participant is still within the 95% confidence interval of the limits of agreement.

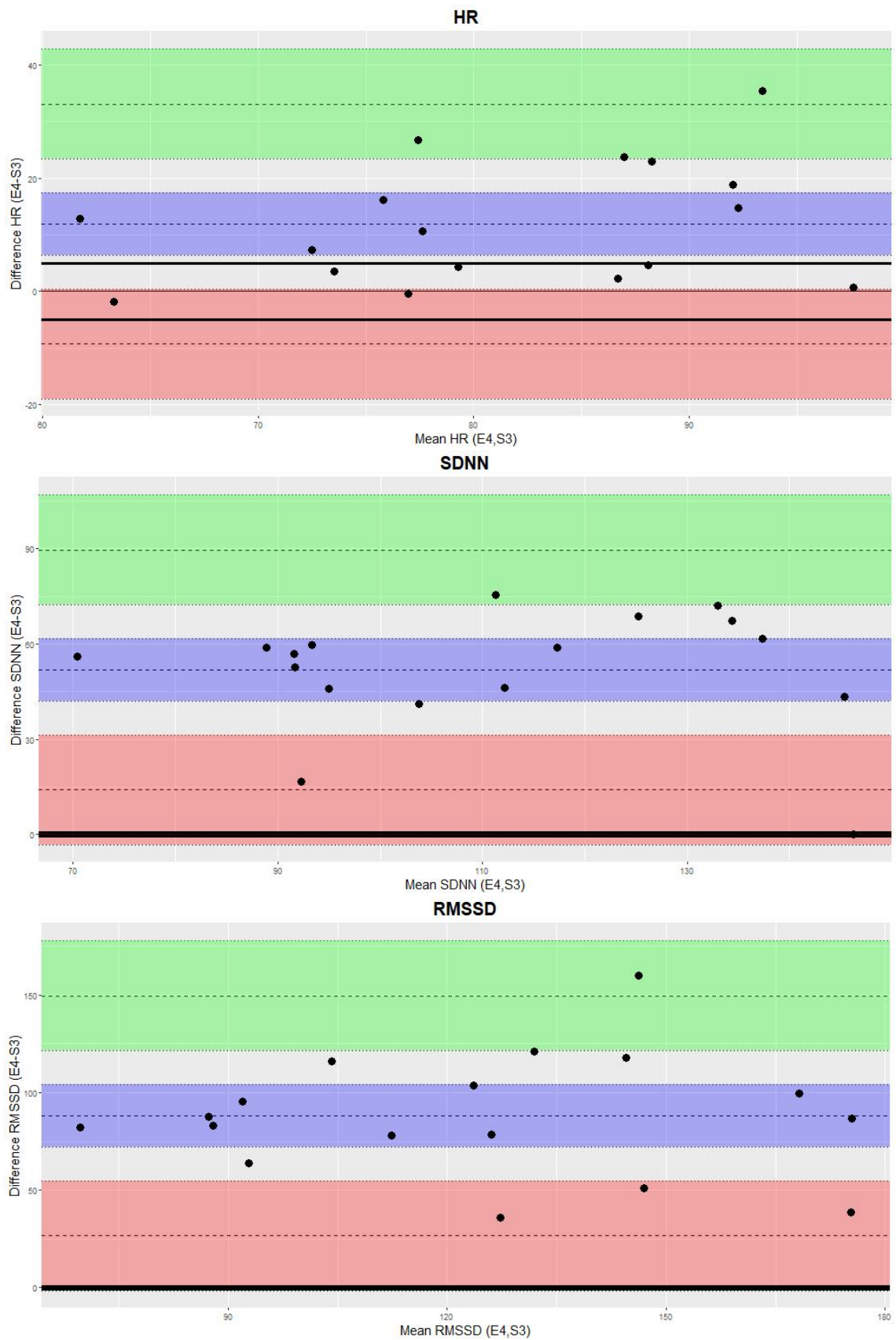
Event Comparison

For the creation of the error bar plots, only a few participants could be used, due to the requirement of having high quality data for three consecutive phases (e.g., BS1, ST1, RC1). In order to include as many participants as possible for the event comparison, the experiment was split up into two parts, consisting of a baseline, either the modified SSST or Stroop task and the related recovery phase. In particular, nine participants were eligible for the first three phases, whereas only five participants were included for the last three phases (see Figure 6).

With the exception of one participant who showed differences in the HR of around 60 bpm for the baseline, as well as recovery phase of the modified SSST, all participants' data lie within the predefined boundaries. However, no clear pattern could be established. For some participants, the modified SSST resulted in a decrease of HR, while for others it increased. The average shows a slight decrease in HR for ST1. With regard to the error bars, only a few participants lie within the error bars which is mainly due to the outlier described previously, resulting in greater error bars for BS1 and RC1, in comparison to ST1.

A clear pattern was found for the Stroop task. Specifically, the HR increased after the BS2 and during ST2 and decreased afterwards (RC2). This trend was even present in the outlying participant who showed differences in HR between E4 and S3 of around 40 bpm at BS2 and 50 bpm at ST2. Most of the participants, as well as the error bar lie within the predefined boundaries at BS2. This is mainly due to the outlier who showed enormous differences in HR among the two sensors. The error bar for ST2 lies outside the defined boundaries, with two participants within the error bar. In contrast, all participants, as well as error bar, are within the boundaries at RC2, displaying differences of less than four bpm.

VALIDATION OF A WRIST-WORN PPG SENSOR

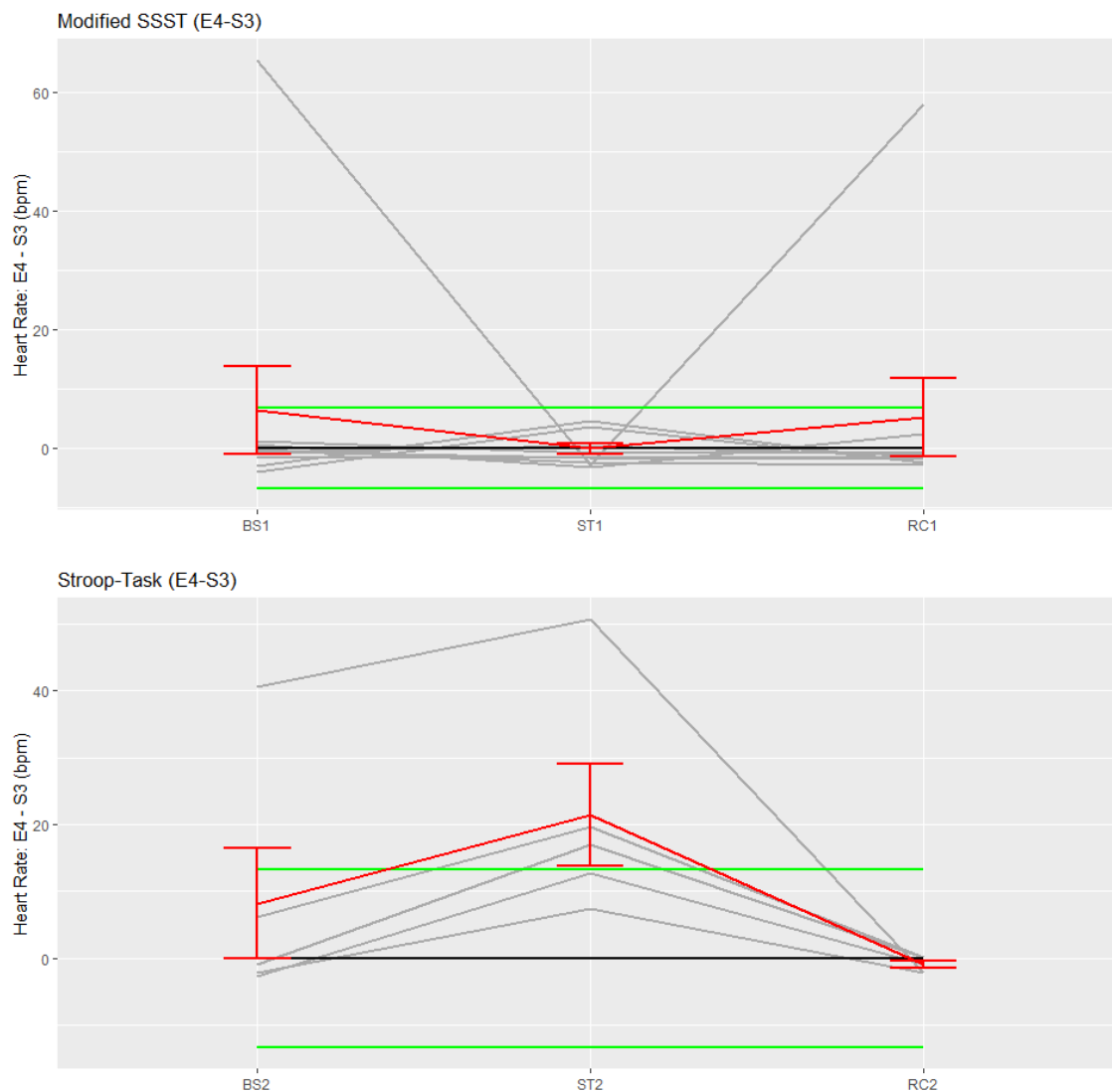
Figure 5*Bland-Altman plots for HR, SDNN and RMSSD*

VALIDATION OF A WRIST-WORN PPG SENSOR

Note. The average of the E4 and S3 is displayed on the horizontal axis and the difference between the two sensors along the vertical axis for heart rate (HR), standard deviation of inter-beat interval (SDNN) and root mean square of successive differences (RMSSD). Each dot represents a participant. Thereby, the blue dotted line shows the mean difference with the purple colored area representing the 95% confidence interval of the mean difference. The red and green dotted lines display the limits of agreement with the colored area around them representing the 95% confidence interval for the limits of agreement. The two bold black lines are the a priori chosen acceptable boundaries.

Figure 6

Error bar plot for HR of each participant for the modified SSST and Stroop task



Note. The error bar plot displays differences in heart rate (HR) for each participant (grey lines) during the modified SSST and the Stroop task with the associated baseline and recovery phase. The red lines show the means and standard errors (SEs), while the black line is the zero axis. The green lines represent the a priori defined boundaries (size of the reference effect).

Discussion

By applying the validity assessment protocol developed by van Lier et al. (2019), the validity of the wrist-worn E4 PPG sensor was assessed by comparing it to the fingertip-worn Shimmer3 PPG sensor. Thereby, the PPG signal was investigated at the signal, parameter and

VALIDATION OF A WRIST-WORN PPG SENSOR

event level by inducing different intensities of stress, induced by the modified SSST and the Stroop task.

The analysis of the subjective stress measurement found a significant difference before and after both stress tasks. In particular, there was an increase in stress for both tasks. Through the analysis at the signal level, no correlations could be found in the PPG measurements of the E4 and S3, implying a low signal level validity. At the parameter level, all participants were within the 95% confidence interval of the limits of agreement. More than half of the measurements were within the predefined boundaries for HR. However, for SDNN and RMSSD no values were within the boundaries. The error bar plot showed a clear pattern for the Stroop task, but not for the modified SSST. With the exception of one participant, all others were in the predefined boundaries with regard to the modified SSST. With regard to the Stroop task, the clear pattern showed an increase in HR differences at ST2, while most participants were within the boundaries.

Subjective Stress Measurement

The significant difference found by the paired sample t-test and the visualization of the average scores show a successful stress induction for the modified SSST, as well as the Stroop task. This indicates that although the increase in stress is not as steep as demonstrated by numerous other studies, both stress tasks led to a significant increase in subjective stress (Dickerson & Kemeny, 2004; Häusser et al., 2012; Kudielka et al., 2004; Linden et al., 1997; von Dawans et al., 2011). With regard to the Stroop task, the results are in line with the literature which found that the Stroop task was effective in inducing psychological stress (Choi et al., 2010; Egilmez et al., 2017; Manuck et al., 1996; Muldoon et al., 1992; Poh et al., 2010). Although the modified SSST showed a significant increase in subjective stress, it was slightly lower than one would expect based on previous research which identified the social stressor to be the most effective (Dickerson & Kemeny, 2004; Egilmez et al., 2017; Linden et al., 1997). One possible explanation for the lower increase of subjective stress by the modified SSST could be the modification itself. Thereby, participants only had to prepare to sing a song, without actually singing out loud, based on previous studies which demonstrated a significant increase of HR, as well as subjective stress measurements during the preparation phase alone (Häusser et al., 2012; van Dick & Mojzisch, 2012; von Drawans et al., 2011). However, the results of the present study indicate that this modified version of the SSST is not strong enough as a stressor to induce the anticipated amount of subjective stress, as only the singing part was left out for the modified SSST.

Signal Level

In this study, the E4 showed a low signal validity, indicating that there was no linear relationship between the measurements of the E4 and the S3 and, consequently, disabling an extensive comparison at the signal level. Although the criterion for validity at the signal level was .80, only cross-correlation coefficients around zero could be found, which for the most part do not reflect the findings of previous research. The majority of studies report good correlations between PPG signals recorded from finger and wrist, whereby the shape of the PPG signal from the finger was described as sharper and more reliable than the wrist (Keikhosravi & Zahedi, 2012; Nardelli et al., 2020; Tăuțan et al., 2015). Besides, Menghini et al. (2019) conducted a study validating the E4 and found similar weak correlations for HR, SDNN and RMSSD during the social stressor and the Stroop task. However, they achieved high correlations for seated and resting phases which are similar to the baseline and recovery phases of the present experiment.

In order to investigate whether the absence of correlations between the E4 and the S3 could be attributed to the data analysis, various methods and approaches were used. Specifically, different downsampling methods were used, namely interpolation, numpy's interpolation, FFT for the Fourier method and poly, provided by the *NeuroKit2* toolbox for python (Makowski et al., 2021). Those different downsampling methods were also tried with an upsampling approach, whereby the E4 was upsampled instead of downsampling the S3. Furthermore, it was controlled for different time lags. However, the individual downsampling methods, the upsampling approach, as well as the time lags yielded lower correlations or around the same correlations, but not higher correlations across all of the relevant phases. Finally, the timestamps which had to be computed for the E4 based on the starting timestamp were recalculated but showed no difference between the original and recalculated timestamps. It was also checked whether the timestamps and lengths of the relevant phases were matching between the two sensors. Overall, these investigations ruled out errors in the data analysis and indicated that the error could be attributed to a systematic error within the sensor, most likely the algorithm of the E4. In particular, the algorithm adjusts the PPG signal, in order to remove artifacts and will be discussed as a possible cause for the absence of correlations between the sensors in the following.

The absence of a linear relationship between the PPG signal of the E4 and the S3, can partially be explained by several differences with regard to the measurement approach. The S3 sensor contains a bright green LED and an ambient light sensor which provides a voltage that is converted by the Shimmer ADC to a 12-bit number that represents the PPG signal in micro Volt (Shimmer, 2016). On the other hand, the E4 sensor uses green and red light and combines

VALIDATION OF A WRIST-WORN PPG SENSOR

them with an algorithm, in order to maximize detection of the pulse wave, according to the website of Empatica (Utilizing the PPG/BVP signal, 2020). Therefore, the measured light contains most information on the pulse wave during exposure to the green LED, whereas the measured light during exposure to the red light contains a reference level to cancel out motion artifacts. However, no specific information could be found regarding the artifact removal algorithm itself and how it selects and processes missing or corrupted data (Ollander et al., 2016). This could include limitations with regard to the validity, due to possible interferences and adjustments by the algorithm which are impossible to detect afterwards. Nevertheless, a linear relationship between the E4 and the S3 was expected. The absence of the linear relationship indicates that the results of the E4 should be interpreted cautiously, especially with regard to the validation of the sensor where accurate measurements are essential.

Furthermore, the difference in approach of obtaining the PPG signal implies different measurement units. Despite the different measurement units, a linear relationship was expected, due to the characteristic parameters of a PPG signal which are present in both signals. However, as Figure 4 indicates, the S3 measures more accurately than the E4 sensor. Specifically, the S3 measured in micro Volt which relates to voltage, whereas the E4 measured in nano Watt which relates to power. The S3 measured values around 1.000 to 2.000 micro Volt, in contrast to the E4 which measured values around the zero axis with positive and negative values in nano Watt. This difference of the measurement units is visualized in Figure 4, where especially the difference with regard to the amplitude of the systolic peak is clearly visible. In particular, this difference in amplitude can be explained by the different measurement sites and their effects on the pulse wave characteristics which attributes a more identifiable waveform characteristic to measurements at the finger, in contrast to recordings at the wrist (Hartmann et al., 2019; Rajala et al., 2018). Furthermore, this includes higher amplitudes of PPG signals at peripheral areas, especially signals recorded at the finger, which are due to larger vascular beds (Hartmann et al., 2019; Maeda et al., 2011; Rajala et al., 2018).

Beside the impact of the sensor placement on waveform characteristics, the measurement site influences the quality of the recorded data. Specifically, sensors at the fingertip were found, in prior research, to measure psychophysiological data more accurately, in contrast to sensors at the wrist which were found to have the highest error, although this was controlled for in the present study by minimizing the movements of the participant and by reminding the participant to move their non-dominant hand with the sensors as little as possible (Longmore et al., 2019; van Dooren et al., 2012). This is partly due to the higher probability of motion artifacts when measuring at the wrist because the sensors need a constant contact with

VALIDATION OF A WRIST-WORN PPG SENSOR

the skin, in order to provide high quality recordings. Therefore, motions like twisting the wrist or just moving the arm around have the potential of creating motion artifacts. With regard to the correlations between both sensors, the validity assessment protocol states that “signals are not expected to have high cross correlation due to placement differences and usage of alternative techniques” (van Lier et al., 2019, p.19). Therefore, these findings do not necessarily imply invalidity of the sensor but emphasize the importance of the investigation of the parameter and event levels.

Parameter Level

The results of the parameter level analysis showed that more than half of the participants were in the predefined boundaries. This is slightly different from studies of Ollander et al. (2016), as well as of Zheng and Poon (2016) who showed that HR of the E4 could be validly determined within the boundaries. Although all values were within the 95% confidence interval of the limits of agreement, the Bland-Alman plot showed that greater averages of HR from E4 and S3 resulted in a greater difference in HR between both sensors. In particular, participants had a difference from up to 20 bpm, with one participant displaying a difference of almost 40 bpm. Therefore, the HR can only be determined validly for averages up to 80 bpm. For greater averages which likely will occur when inducing stress, the HR measurement of the E4 cannot be validly determined. With regard to the SDNN, as well as RMSSD, no values were within the predefined boundaries. This could be attributed to a systematic bias which is caused by the different ways of measuring the PPG signal or the need for a different way of analyses as the signals are in principle quite different, reflected by greater E4 values for HR, SDNN and RMSSD. The differences in the extracted parameters were shown to be large and led to values within the limits of agreement, while being outside the predefined boundaries. Therefore, the HR of the E4 can be determined validly only for lower HRs, whereas the SDNN and RMSSD cannot be determined validly, due to the large difference in the extracted parameters.

Event Level

In contrast to the subjective stress measurements, the physiological measurements could not replicate the increase in measured stress. The analysis at the event level showed no clear pattern for the modified SSST. However, except for one participant, all others were in the predefined boundaries. This outlying participant is the reason for the greater height of the error bars at BS1 and RC1, due to differences in HR of around 60 bpm. With regard to the Stroop task, the error bar plot showed that the values for BS2 and RC2 were within the boundaries, whereas most participants were outside the predefined boundaries for ST2. Taking into account figure 3 which displays the HR per participant for each sensor and task, allows for a more

VALIDATION OF A WRIST-WORN PPG SENSOR

detailed discussion. In particular, the E4 was not able to detect an increase in HR for the modified SSST, whereas for the Stroop task, the S3 did not measure an increase. Therefore, the results at the event level are inconclusive for both, the modified SSST and the Stroop task. Furthermore, it can be concluded that both tasks were not strong enough stressors to elevate the HR and to see differences in the physiological arousal.

This is in contrast to the literature which found social and cognitive stressors to be the most effective in inducing stress at the HR level (Choi et al., 2010; Egilmez et al., 2017; Manuck et al., 1996; Muldoon et al., 1992; Poh et al., 2010). With regard to the modified SSST, studies demonstrated that the HR response pattern significantly increased during the preparation phase (Kudielka et al., 2004; von Dawans et al., 2011). However, the responses measured by the E4 could not be distinguished from each other. This could be due to the short stressor and the absence of the singing. Several studies found the wrist to be only sensitive to larger stressors (Ollander et al., 2016; van Dooren et al., 2012). With regard to the Stroop task, the responses measured by the S3 could not be distinguished from each other. A possible explanation could be that participants moved too much during the task, although the instructions were to only use the dominant hand and hold the non-dominant hand with the sensors as still as possible, as the S3 was found to be extremely sensitive toward physical exercise (Longmore et al., 2019). Another possible explanation for the inconclusive results could be that the difficult level of the Stroop task did not change. This way, the induced stress decreased rather quickly, due to a habituation effect. By implementing an adaptation algorithm which decreases response times, the Stroop task could be made gradually more difficult even for participants who performed well during the first phases of the task, as well as create a steady stress inducement which then can be evaluated by examining time on task effects.

With regard to the subjective measurements which could not be replicated by the physiological measurements, the findings are supported by a study of Campbell and Ehlert (2012) who investigated the relationship between the physiological responses induced by a social stressor such as the SSST and subjective experience and found non-existent to moderate correlations. However, this only would explain the results of the modified SSST and not of the Stroop task, indicating that both tasks are not strong enough stressors.

Beside the fact that the stressors were not strong enough, there was too much data loss of the E4. Specifically, three participants were removed completely, due to unrealistic values measured by the E4. After applying the SQI, 37% of the data was removed which led to the removal of another three participants, resulting in only 14 participants for the analysis. The removal included each individual phase which did not have an SQI of at least 70. For the event

VALIDATION OF A WRIST-WORN PPG SENSOR

level analysis, participants were required to have high quality data for three consecutive phases (e.g., BS1, ST1, RC1), whereby such a phase had to be excluded if an individual phase was removed due to insufficient quality. In case BS1 of a participant was removed due to its SQI of 40, all three consecutive phases had to be removed for both sensors which resulted in only nine participants for the first three phases and five participants for the last three phases. This led to the exclusion of S3 data which were otherwise applicable for the analysis of the modified SSST, as well as the Stroop task. With regard to the data loss, it is possible that the algorithm of the E4 which removes artifacts could have played a role, due to the fact that it is not possible to detect where the algorithm adjusted measurements or even how it works (Ollander et al., 2016). Nevertheless, the data loss of the E4 is supported by Longmore et al. (2019) who found that the E4 had the highest error overall, compared to other sensors. It can thus be concluded that the E4 sensor does not measure stress as accurately compared to the S3. Furthermore, the amount of excluded data of the E4 suggests that the algorithm which removes artifacts or the E4 itself is not as sensitive towards weak stressors as the S3 and therefore should not be used in validation studies or studies which do not use strong stressors.

Limitations

One limitation of the present study was the exclusion of data that did not meet the prescribed quality criteria. This limits the generalization of the findings with regard to the validity, especially with the goal being the validation of a biosensor. Almost 40% of the PPG data had to be excluded, whereby the sample size amplified the problem, resulting in only a few participants who could actually be included in the analysis. Although the S3 measured more accurately, many parts of the recorded S3 data had to be excluded, due to the low quality of the E4. Therefore, future research should investigate the influence of the E4's algorithm on the artifact removal and quality of the recorded PPG signal. Furthermore, it is suggested to examine different analysis methods for the signal level evaluation of the E4 and S3, due to the different waveforms and measurement units of the recordings. Based on the demonstrated low validity and inconclusive results, it is advised to deliberately consider the context in which to use the E4. Specifically, the E4 should only be used in ambulatory assessments for the investigation of strong and long-lasting stressors. A validated alternative should be used if the aim is to measure subtle variations in HR parameters or to make meaningful decisions regarding a person's health (e.g., medical- / treatment-decisions). For those reasons, it is suggested to apply the SQI beforehand in a pilot trial, in order to ensure that a sensor produces data with a certain quality. In case of the E4, the application of the SQI would have showed that a large part of the data did not meet the required quality standards for a validation study.

VALIDATION OF A WRIST-WORN PPG SENSOR

Another limitation was the physiological stress induction which was not successful and also contributed to the inconclusive results with regard to the event level, resulting in indistinguishable signals of the E4. With regard to the SSST, a shorter version of the SSST was used, in order to be more time efficient. However, as the present study showed, the stressor was not strong enough to induce enough stress in the participants. This issue can be addressed by using longer and more intensive social stressors (e.g., original SSST, Trier Social Stress Test), consequently, increasing the overall length of the experiment. With regard to the Stroop task, the level of difficulty should be increased over time by adapting the presentation time, as well as by introducing more or other words and colors, respectively. Further studies are needed, in order to investigate whether the reason for not detecting the physiological arousal is due to the sensitivity of the stressor or the sensor itself.

Conclusion

The application of the validity assessment protocol showed that the validity of the E4 is lower than described in the literature (McCarthy et al., 2016). Specifically, the successful induction of subjective stress for both the modified SSST and the Stroop task could not be replicated by the physiological measurements. Although the present study was unable to make a definitive statement about the validity of the E4, the approach of taking a PPG sensor as RD, as well as the validity assessment protocol itself showed its potential. Instead of an ECG, the PPG sensors are more available, less time-consuming and less restricting with regard to participants and setting. However, it is important for future research to find applicable PPG sensors as reference devices, with regard to artifact sensitivity, the sensor placement and the technical measurement approach of the sensor. Furthermore, stronger and more long-lasting stressors should be used, in order to detect significant increases in HR arousal and possible patterns. This way, it can be examined if the inability to accurately measure the stress induction is due to the stressor or the sensor itself. By implementing those recommendations, data loss to the extent of the present study should be prevented, enabling the extraction and analysis of parameters, as well as the determination of validity.

References

- Abbod, M., Chiou, Y. R., Yang, S. H., Fan, S. Z., & Shieh, J. S. (2011). Developing a monitoring psychological stress index system via photoplethysmography. *Artificial Life and Robotics*, 16(3), 430-433, <https://doi.org/10.1007/s10015-011-0976-y>
- Akhter, N., Tharewal, S., Kale, V., Bhalerao, A., & Kale, K. V. (2016). Heart-based biometrics and possible use of heart rate variability in biometric recognition systems. In R. Chaki, A. Cortesi, K. Saeed, & N. Chaki (eds), *Advanced Computing and Systems for Security. Advances in Intelligent Systems and Computing*, Vol. 395 (pp. 15-29). India: Springer.
- Al'Absi, M., Bongard, S., Buchanan, T., Pincomb, G. A., Licinio, J., & Lovallo, W. R. (1997). Cardiovascular and neuroendocrine adjustment to public speaking and mental arithmetic stressors. *Psychophysiology*, 34(3), 266-275.
- Allen, J. (2007). Photoplethysmography and its application in clinical physiological measurement. *Physiological Measurement*, 28(3), 1-39, <https://doi.org/10.1088/0967-3334/28/3/R01>
- Appel, M. L., Berger, R. D., Saul, J. P., Smith, J. M., & Cohen, R. J. (1989). Beat to beat variability in cardiovascular variables: Noise or music? *Journal of the American College of Cardiology*, 14(5), 1139-1148, [https://doi.org/10.1016/0735-1097\(89\)90408-7](https://doi.org/10.1016/0735-1097(89)90408-7)
- Baum, A., & Contrada, R. (Eds.). (2010). *The handbook of stress science: Biology, psychology, and health*. New York: Springer Publishing Company.
- Bolanos, M., Nazeran, H., & Haltiwanger, E. (2006, August). *Comparison of heart rate variability signal features derived from electrocardiography and photoplethysmography in healthy individuals*. Paper presented at the 28th IEEE EMBS Annual International Conference, New York City.
- Bonato, P. (2003). Wearable sensors/systems and their impact on biomedical engineering. *IEEE Engineering in Medicine and Biology Magazine*, 22(3), 18-20, <https://doi.org/10.1109/MEMB.2003.1213622>
- Brouwer, A.-M., & Hogervorst, M. A. (2014). A new paradigm to induce mental stress: The sing-a-song stress test (SSST). *Frontiers in Neuroscience*, 8, 224-231, <https://doi.org/10.3389/fnins.2014.00224>

VALIDATION OF A WRIST-WORN PPG SENSOR

- Campbell, J., & Ehler, U. (2012). Acute psychosocial stress: Does the emotional stress response correspond with physiological responses? *Psychoneuroendocrinology*, 37(8), 1111-1134, <https://doi.org/10.1016/j.psyneuen.2011.12.010>
- Choi, J., Ahmed, B., & Gutierrez-Osuna, R. (2010). *Ambulatory stress monitoring with minimally-invasive wearable sensors* (Technical Report tamu-cs-tr-2010-11-1). Department of Computer Science and Engineering, Texas A&M University.
- Chrousos, G. P. (2009). Stress and disorders of the stress system. *Nature Reviews Endocrinology*, 5(7), 374-381, <https://doi.org/10.1038/nrendo.2009.106>
- Cook, S., Togni, M., Schaub, M. C., Wenaweser, P., & Hess, O. M. (2006). High heart rate: A cardiovascular risk factor? *European Heart Journal*, 26, 2387-2393, <https://doi.org/10.1093/eurheartj/ehl259>
- Dickerson, S. S., & Kemeny, M. E. (2004). Acute stressors and cortisol responses: A theoretical integration and synthesis of laboratory research. *Psychological Bulletin*, 130(3), 355-391, <https://doi.org/10.1037/0033-2909.130.3.355>
- Egilmez, B., Poyraz, E., Zhou, W., Memik, G., Dinda, P., & Alshurafa, N. (2017, March 13-17). *UStress: Understanding college student subjective stress using writs-based passive sensing*. Paper presented at the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 673-678, Kona, USA, <https://doi.org/10.1109/PERCOMW.2017.7917644>
- Elgendi, M. (2012). On the analysis of fingertip photoplethysmogram signals. *Current Cardiology Reviews*, 8, 14-25, <https://doi.org/10.2174/157340312801215782>
- Elgendi, M., Jonkman, M., & DeBoer, F. (2010, January). Heart rate variability and the acceleration plethysmogram signals measured at rest. In A. Fred, J. Filipe & H. Gamboa (eds.), *International Joint Conference on Biomedical Engineering Systems and Technologies* (pp. 266-277). Valencia, Spain: Springer.
- Fletcher, R. R., Tam, S., Omojola, O., Redemske, R., Redor, S., & Moshoka, J. M. (2011, May 23-26). *Mobile application and wearable sensors for use in cognitive behavioral therapy for drug addiction and PTSD*. Paper presented at the Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference, Dublin, <https://doi.org/10.4108/icst.pervasivehealth.2011.246105>

VALIDATION OF A WRIST-WORN PPG SENSOR

- Fusco, A., Locatelli, D., Onorati, F., Durelli, G. C., & Santambrogio, M. D. (2015, October). *On how to extract breathing rate from PPG signal using wearable devices*. In Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE (pp. 1-4). Atlanta, GA: IEEE.
- Giardino, N. D., Lehrer, P. M., & Edelberg, R. (2002). Comparison of finger plethysmograph to ECG in the measurement of heart rate variability. *Psychophysiology*, 39, 246-253, <https://doi.org/10.1017.S0048577201392168>
- Gil, E., Orini, M., Bailón, R., Vergara, J. M., Mainardi, L., & Laguna, P. (2010). Photoplethysmography pulse rate variability as a surrogate measurement of heart rate variability during non-stationary conditions. *Physiological Measurement*, 31, 1271-1290, <https://doi.org/10.1088/0967-3334/31/9/015>
- Goldsack, J. C., Coravos, A., Bakker, J. P., Bent, B., Dowling, A. V., Fitzer-Attas, C., Godfrey, A., Godino, J. G., Gujar, N., Izmailova, E., Manta, C., Peterson, B., Vandendriessche, B., Wood, W. A., Wang, K. W., & Dunn, J. (2020). Verification, analytical validation, and clinical validation (V3): The foundation of determining fit-for-purpose for biometric monitoring technologies (BioMeTs). *npj digital Medicine*, 3(1), 1-15, <https://doi.org/10.1038/s41746-020-0260-4>
- Greve, M., Kviesis-Kipge, E., Rubenis, O., Rubenis, U., Mecnika, V., Grabovskis, A., & Marcinkevics, Z. (2012, April). *Comparison of pulse rate variability derived from digital photoplethysmography over the temporal artery with the heart rate variability derived from a polar heart rate monitor*. Paper presented at the 7th ESGCO, Kazimierz Dolny.
- Groth-Marnat, G. (2003). *Handbook of psychological assessment (4th Edition)*. Hoboken, New Jersey: John Wiley & Sons Inc.
- Gu, W. B., Poon, C. C. Y., Sy, M. Y., Leung, H. K., Liang, Y. P., & Zhang, Y. T. (2009, June 3-5). *A t-shirt-based body sensor network for cuffless calibration and estimation of arterial blood pressure*. Paper presented at the Wearable and Implantable Body Sensor Networks, 6th International Workshop, Berkeley, <https://doi.org/10.1109/BSN.2009.23>
- Häusser, J. A., Kattenstroth, M., van Dick, R., & Mojzisch, A. (2012). “We” are not stressed: Social identity in groups buffers neuroendocrine stress reactions. *Journal of*

VALIDATION OF A WRIST-WORN PPG SENSOR

- Experimental Social Psychology*, 48(4), 973-977,
<https://doi.org/10.1016/j.jesp.2012.02.020>
- Hartmann, V., Liu, H., Chen, F., Qiu, Q., Hughes, S., & Zheng, D. (2019). Quantitative comparison of photoplethysmographic waveform characteristics: Effect of measurement site. *Frontiers in Physiology*, 10, 198-205,
<https://doi.org/10.3389/fphys.2019.00198>
- Jeyhani, V., Mahdiani, S., Peltokangas, M., & Vehkaoja, A. (2015). *Comparison of HRV parameters derived from photoplethysmography and electrocardiography signals*. Paper presented at the 37th IEEE Engineering in Medicine and Biology Society Conference, Milan, <https://doi.org/10.1109/EMBC.2015.7319747>
- Jo, E., Lewis, K., Directo, D., Kim, M. J., & Dolezal, B. A. (2016). Validation of biofeedback wearables for photoplethysmographic heart rate tracking. *Journal of Sports Science and Medicine*, 15, 540-547.
- Julian, L. J. (2011). Measures of anxiety - State-trait anxiety inventory (STAI), beck anxiety inventory (BAI), and hospital anxiety and depression scale-anxiety (HADS-A). *Arthritis Care & Research*, 63(11), 467-472, <https://doi.org/10.1002/acr.20561>
- Kamarck, T. W., & Lovallo, W. R. (2003). Cardiovascular reactivity to psychological challenge: conceptual and measurement considerations. *Psychosomatic Medicine*, 65(1), 9-21, <https://doi.org/10.1097/01.psy.0000030390.34416.3e>.
- Karlen, W., Kobayashi, K., Ansermino, J. M., & Dumont, G. A. (2012). Photoplethysmogram signal quality estimation using repeated gaussian filters and cross-correlation. *Physiological measurement*, 33(10), 1617-1629, <https://doi.org/10.1088/0967-3334/33/10/1617>
- Keikhosravi, A., & Zahedi, E. (2012, December). Effect of deep breath on the correlation between wrist and finger photoplethysmograms. In *2012 19th Iranian Conference of Biomedical Engineering (ICBME)* (pp. 135-138). IEEE,
<https://doi.org/10.1109/ICBME.2012.6519672>
- Kemeny, M. E. (2003). The psychobiology of stress. *Current directions in psychological science*, 12(4), 124-129, <https://doi.org/10.1111/1467-8721.01246>

- Kirschbaum, C., Pirke, K. M., & Hellhammer, D. H. (1993). The 'trier social stress test'—A tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology*, 28(1-2), 76-81, <https://doi.org/10.1159/000119004>
- Kudielka, B. M., Buske-Kirschbaum, A., Hellhammer, D. H., & Kirschbaum, C. (2004). Differential heart rate reactivity and recovery after psychosocial stress (TSST) in healthy children, younger adults, and elderly adults: The impact of age and gender. *International Journal of Behavioral Medicine*, 11(2), 116-121, https://doi.org/10.1207/s15327558ijbm1102_8
- Kvaal, K., Ulstein, I., Nordhus, I. H., & Engedal, K. (2005). The spielberger state-trait anxiety inventory (STAI): The state scale in detecting mental disorders in geriatric patients. *International Journal of Geriatric Psychiatry*, 20, 629-634, <https://doi.org/10.1002/gps.1330>
- Lehnert, B. (2015). *BlantAltmanLeh*. R package version 0.3.1, URL: <https://CRAN.R-project.org/package=BlandAltmanLeh>.
- Lin, W.-H., Wu, D., Li, C., Zhang, H., & Zhang, Y.-T. (2014). Comparison of heart rate variability from PPG with that from ECG. In Y.-T. Zhang (eds), *The International Conference on Health Informatics. IFMBE Proceedings*, Vol. 42 (pp. 213-215). Cham: Springer, https://doi.org/10.1007/978-3-319-03005-0_54
- Longmore, S. K., Lui, G. Y., Naik, G., Breen, P. P., Jalaludin, B., & Gargiulo, G. D. (2019). A comparison of reflective photoplethysmography for detection of heart rate, blood oxygen saturation, and respiration rate at various anatomical locations. *Sensors*, 19(8), 1874.
- Maeda, Y., Sekine, M., & Tamura, T. (2011). Relationship between measurement site and motion artifacts in wearable reflected photoplethysmography. *Journal of Medical Systems*, 35(5), 969-976, <https://doi.org/10.1007/s10916-010-9505-0>
- Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., Schölzel, C., & Chen, S. A. (2021). NeuroKit2: A python toolbox for neurophysiological signal processing. *Behavior Research Methods*, 53, 1689-1696, <https://doi.org/10.3758/s13428-020-01516-y>

VALIDATION OF A WRIST-WORN PPG SENSOR

- Mandryk, R. L., Inkuben, K. M., & Calvert, T. W. (2006). Using psychophysiological techniques to measure user experience with entertainment technologies. *Behavior & Information Technology*, 25(2), 141-158, <https://doi.org/10.1080/01449290500331156>
- Manuck, S. B., Polefrone, J. M., Terrell, D. F., Muldoon, M. F., Kasprovicz, A. L., Waldstein, S. R., Jennings, R. J., Malkoff, S. B., Marsland, A., & Graham, R. E. (1996). Absence of enhanced sympathoadrenal activity and behaviorally evoked cardiovascular reactivity among offspring of hypertensives. *American Journal of Hypertension*, 9(3), 248-255.
- Marteau, T. M., & Bekker, H. (1992). The development of a six-item short-form of the state scale of the spielberger state-trait anxiety inventory (STAI). *British Journal of Clinical Psychology*, 31, 301-306.
- McCarthy, C., Pradhan, N., Redpath, C., & Adler, A. (2016, May). Validation of the Empatica E4 wristband. In *Student Conference (ISC), 2016 IEEE EMBS International* (pp. 1-4). Ottawa: IEEE.
- Menghini, L., Gianfranchi, E., Cellini, N., Patron, E., Tagliabue, M., & Sarlo, M. (2019). Stressing the accuracy: Wrist-worn wearable sensor validation over different conditions. *Psychophysiology*, 56(11), 13441-13456, <https://doi.org/10.1111/psyp.13441>
- Mitsuhashi, R., Iuchi, K., Goto, T., Matsubara, A., Hirayama, T., Hashizume, H., & Tsumura, N. (2019, July). Video-based stress level measurement using imaging photoplethysmography. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (pp. 90-95). IEEE, <https://doi.org/10.1109/ICMEW.2019.0-105>.
- Muldoon, M. F., Bachen, E. A., Manuck, S. B., Waldstein, S. R., Bricker, P. L., & Bennett, J. A. (1992). Acute cholesterol responses to mental stress and change in posture. *Archives of Internal Medicine*, 152(4), 775-780.
- Mullan, P., Kanzler, C. M., Lorch, B., Schroeder, L., Winkler, L., Laich, L., Riedel, F., Richer, R., Luckner, C., Leutheuser, H., Eskofier, B. M., & Pasluosta, C. (2015, August). Unobtrusive heart rate estimation during physical exercise using photoplethysmographic and acceleration data. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE* (pp. 6114-6117). Milan, Italy: IEEE.

VALIDATION OF A WRIST-WORN PPG SENSOR

- Munos, B., Baker, P. C., Bot, B. M., Crouthamel, M., de Vries, G., Ferguson, I., Hixon, J. D., Malek, L. A., Mastrototaro, J. J., Misra, V., Ozcan, A., Sacks, L., & Wang, P. (2016). Mobile health: The power of wearables, sensors, and apps to transform clinical trials. *Annals of the New York Academy of Sciences*, 1375(1), 3-18, <https://doi.org/10.1111/nyas.13117>
- Nardelli, M., Vanello, N., Galperti, G., Greco, A., & Scilingo, E. P. (2020). Assessing the Quality of heart rate variability estimated from wrist and finger PPG: A novel approach based on cross-mapping method. *Sensors*, 20(11), 3156-3168, <https://doi.org/10.3390/s20113156>
- Nath, R. K., Thapliyal, H., & Caban-Holt, A. (2020, January). Validating physiological stress detection model using cortisol as stress bio marker. In *2020 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 1-5). IEEE.
- Ollander, S., Godin, C., Campagne, A., & Charbonnier, S. (2016, October). A comparison of wearable and stationary sensors for stress detection. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 004362-004366). Budapest, Hungary: IEEE.
- Orphanidou, C., Bonnici, T., Charlton, P., Clifton, D., Vallance, D., & Tarassenko, L. (2014). Signal-quality indices for the electrocardiogram and photoplethysmogram: Derivation and applications to wireless monitoring. *IEEE Journal of Biomedical and Health Informatics*, 19(3), 832-838, <https://doi.org/10.1109/JBHI.2014.2338351>
- Quintana, D. S., & Heathers, J. A. (2014). Considerations in the assessment of heart rate variability in biobehavioral research. *Frontiers in Psychology*, 5, 78-87, <https://doi.org/10.3389/fpsyg.2014.00805>.
- Parasnis, R., Pawar, A., & Manivannan, M. (2015, November). Multiscale entropy and poincare plot-based analysis of pulse rate variability and heart rate variability of ICU patients. In *International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS) 2015* (pp. 290-295). Okinawa, Japan: IEEE.
- Peirce, J. W., Gray, J. R., Simpson, S., MacAskill, M. R., Höchenberger, R., Sogo, H., Kastman, E., Lindeløv, J. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, 51(1), 195-203, <https://doi.org/10.3758/s13428-018-01193-y>

VALIDATION OF A WRIST-WORN PPG SENSOR

- Picard, R. W., & Healey, J. (1997). Affective wearables. *Personal Technologies*, 1(4), 231-240, <https://doi.org/10.1007/BF01682026>
- Pietilä, J., Mehrang, S., Tolonen, J., Helander, E., Jimison, H., Pavel, M., & Korhonen, I. (2017). Evaluation of the accuracy and reliability for photoplethysmography based heart rate and beat-to-beat detection during daily activities. In *EMBECE & NBC 2017* (pp. 145-148). Singapore: Springer.
- Poh, M.-Z., Swenson, N. C., & Picard, R. W. (2010). A wearable sensor for unobtrusive, long-term assessment of electrodermal activity. *IEEE Transactions on Biomedical Engineering*, 57(5), 1243-1252, <https://doi.org/10.1109/TBME.2009.2038487>
- Poon, C. C. Y., Wong, Y. M., & Zhang, Y. T. (2006, July). *M-Health: The development of cuff-less and wearable blood pressure meters for use in body sensor networks*. Paper presented at Life Science Systems and Applications Workshop, Bethesda, <https://doi.org/10.1109/LSSA.2006.250377>
- R Core Team (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL: <https://www.R-project.org/>.
- Ragot, M., Martin, N., Em, S., Pallamin, N., & Diverrez, J.-M. (2017). Emotion recognition using physiological signals: Laboratory vs. wearable sensors. In T. Ahram & C. Falcao (eds.), *Advances in human factors in wearable technologies and game design*. AHFE 2017. Advances in Intelligent Systems and Computing, 608, 15-22. Los Angeles, California: Springer, https://doi.org/10.1007/978-3-319-60639-2_2
- Rajala, S., Lindholm, H., & Taipalus, T. (2018). Comparison of photoplethysmogram measured from wrist and finger and the effect of measurement location on pulse arrival time. *Physiological Measurement*, 39(7), 075010-0750129, <https://doi.org/10.1088/1361-6579/aac7ac>
- Santos, S. A., Venema, B., & Leonhardt, S. (2012). Accelerometer-assisted PPG measurement during physical exercise using the LAVIMO sensor system. *Acta Polytechnica*, 52(5), 80-85, <https://doi.org/10.14311/1646>
- Sartor, F., Papini, G., Cox, L. G. E., & Cleland, J. (2018). Methodological shortcomings of wrist-worn heart rate monitors validations. *Journal of Medical Internet Research*, 20(7), 31-36, <https://doi.org/10.2196/10108>

VALIDATION OF A WRIST-WORN PPG SENSOR

- Selvaraj, N. (2015, August). Psychological acute stress measurement using a wireless adhesive biosensor. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 3137-3140). IEEE.
- Selvaraj, N., Jaryal, A., Santhosh, J., Deepak, K. K., & Anand, S. (2008). Assessment of heart rate variability derived from finger-tip photoplethysmography as compared to electrocardiography. *Journal of Medical Engineering & Technology*, 32(6), 479-484, <https://doi.org/10.1080/03091900701781317>
- Shcherbina, A., Mattsson, C. M., Waggott, D., Salisbury, H., Christle, J. W., Hastie, T., Wheeler, M. T., & Ashley, E. A. (2017). Accuracy in wrist-worn, sensor-based measurements of heart rate and energy expenditure in a diverse cohort. *Journal of Personalized Medicine*, 7(2), 3-14, <https://doi.org/10.3390/jpm7020003>
- Shimmer (2016). Optical pulse sensor user guide revision 1.6. Retrieved from https://www.shimmersensing.com/images/uploads/docs/Optical_Pulse_Sensor_User_Guide_rev1.6.pdf. Retrieved at 02.06.2021
- Singh, R. R., Conjeti, S., & Banerjee, R. (2011, October). An approach for real-time stress-trend detection using physiological signals in wearable computing systems for automotive drivers. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (pp. 1477-1482). IEEE.
- Spielberger, C. D., Gorsuch, R. L., Lushene, R., Vagg, P. R., & Jacobs, G. A. (1983). *Manual for the state-trait anxiety inventory*. Palo Alto, CA: Consulting Psychologists Press.
- Tapia, E. M., Intille, S. S., Haskell, W., Larson, K., Wright, J., King, A., & Friedman, R. (2007, October 11-13). *Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor*. Paper presented at Wearable Computers, 2007 11th IEEE International Symposium, Boston, <https://doi.org/10.1109/ISWC.2007.4373774>
- Tăuțan, A. M., Young, A., Wentik, E., & Wieringa, F. (2015, August). Characterization and reduction of motion artifacts in photoplethysmographic signals from a wrist-worn device. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 6146-6149). Milan, Italy: IEEE.

VALIDATION OF A WRIST-WORN PPG SENSOR

- Thuczek, A., Henriques, J. B., & Brown, R. L. (2009). Support for the reliability and validity of a six-item state anxiety scale derived from the state-trait anxiety inventory. *Journal of Nursing Measurement, 17*(1), 19-28.
- Utilizing the PPG/BVP signal. (23.01.2020). Retrieved from <https://support.empatica.com/hc/en-us/articles/204954639-Utilizing-the-PPG-BVP-signal>. Retrieved at 02.06.2021
- van Dooren, M., de Vries, J. J. G. G. J., & Janssen, J. H. (2012). Emotional sweating across the body: Comparing 16 different skin conductance measurement locations. *Physiology and Behavior, 106*(2), 298-304, <https://doi.org/10.1016/j.physbeh.2012.01.020>
- van Lier, H. G., Pieterse, M. E., Garde, A., Postel, M. G., de Haan, H. A., Vollenbroek-Hutten, M. M., Schraagen, J. M. & Noordzij, M. L. (2019). A standardized validity assessment protocol for physiological signals from wearable technology: Methodological underpinnings and an application to the E4 biosensor. *Behavior Research Methods, 52*, 607-629, <https://doi.org/10.3758/s13428-019-01263-9>
- von Dawans, B., Kirschbaum, C., & Heinrichs, M. (2011). The trier social stress test for groups (TSST-G): A new research tool for controlled simultaneous social stress exposure in a group format. *Psychoneuroendocrinology, 36*(4), 514-522, <https://doi.org/10.1016/j.psyneuen.2010.08.004>
- Wander, J. D., & Morris, D. (2014). A combined segmenting and non-segmenting approach to signal quality estimation for ambulatory photoplethysmography. *Physiological Measurement, 35*(12), 2543-2561, <https://doi.org/10.1088/0967-3334/35/12/2543>
- Wilard, R. M., Inan, O. T., Argyres, B., Temadi, M., Kovacs, G. T. A., & Giovannardi, L. (2011). Automatic detection of motion artifacts in the ballistocardiogram measured on a modified bathroom scale. *Medical & Biological Engineering & Computing, 49*(2), 213-220, <https://doi.org/10.1007/s11517-010-0722-y>
- Yoo, K. S., & Lee, W. H. (2011, June). Mental stress assessment based on pulse photoplethysmography. In *2011 IEEE 15th International Symposium on Consumer Electronics (ISCE)* (pp. 323-326). IEEE.
- Zaki, R., Bulgiba, A., Ismail, R., & Ismail, N. A. (2012). Statistical methods used to test for agreement of medical instruments measuring continuous variables in method

VALIDATION OF A WRIST-WORN PPG SENSOR

- comparison studies: A systematic review. *PloS one*, 7(5), e37908, <https://doi.org/10.1371/journal.pone.0037908>
- Zangróniz, R., Martínez-Rodrigo, A., López, M. T., Pastor, J. M., & Fernández-Caballero, A. (2018). Estimation of mental distress from photoplethysmography. *Applied Sciences*, 8(1), 69-83, <https://doi.org/10.3390/app8010069>
- Zhang, Y., Song, S., Vullings, R., Biswas, D., Simões-Capela, N., Van Helleputte, N., van Hoof, C., & Groenendaal, W. (2019). Motion artifact reduction for wrist-worn photoplethysmograph sensors based on different wavelengths. *Sensors*, 19(3), 673-690, <https://doi.org/10.3390/s19030673>
- Zheng, Y., Ding, X., Poon, C. C., Lo, B. P., Zhang, H., Zhou, X., Yang, G., Zhao, N., & Zhang, Y. (2014). Unobtrusive sensing and wearable devices for health informatics. *IEEE Transactions on Biomedical Engineering*, 61(5), 1538-1554, <https://doi.org/10.1109/TBME.2014.2309951>
- Zheng, Y., & Poon, C. C. Y. (2016, May). Wearable devices and their applications in surgical robot control and p-medicine. In *2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 659-663). IEEE, <https://doi.org/10.1109/CSCWD.2016.7566067>

Appendix

Appendix A – Screening Questions

The following questions were asked, in order to examine if potential participants fulfill any of the exclusion criteria.

1. Do you currently suffer from any physical or psychological illness?
2. Do you have any preexisting heart or lung condition?
For Example:
 - a. cardiovascular disease
 - b. pulmonary disease
 - c. arrhythmias
 - d. aorta disease
 - e. coronary artery disease
 - f. hypertension
3. Have you ever been told by a doctor, nurse, or other health professional that you have high blood pressure?
4. Do you currently take any prescribed medication (especially medication affecting the cardiovascular or nervous system)?
5. Do you have any substance addiction (e.g., alcohol, drugs, medications)?

VALIDATION OF A WRIST-WORN PPG SENSOR

Appendix B – Experiment Script (Psychopy)

```

1 #imports
2 import os
3 import sys
4 import csv
5 import time
6 import random
7
8 import numpy as np
9
10 import pygame
11 from pygame.locals import *
12 from pygame.compat import unicr_, unicode_
13
14 import psychopy.gui
15 import psychopy.visual
16 import psychopy.core
17 import psychopy.event
18
19 #set directory
20 os.chdir('/Users/ene/Documents/Python_Scripts')
21
22 #set GUI
23 gui = psychopy.gui.Dlg()
24
25 gui.addField("Subject ID:")
26
27 gui.show()
28
29 subj_id = gui.data[0]
30 print("Subject ID: " + subj_id)
31
32 #check for existing file
33 data_path = "p_" + subj_id + "_time_data.csv"
34
35 if os.path.exists(data_path):
36     sys.exit("Data path " + data_path + " already exists!")
37
38 data_path = "p_" + subj_id + "_stai_data.csv"
39
40 if os.path.exists(data_path):
41     sys.exit("Data path " + data_path + " already exists!")
42
43 data_path = "p_" + subj_id + "_stroop_data.csv"
44
45 if os.path.exists(data_path):
46     sys.exit("Data path " + data_path + " already exists!")
47
48 responses = []
49
50 #variables
51 stai_1 = []
52 stai_2 = []
53 stai_3 = []
54 stai_4 = []
55 stai_5 = []
56 stai_6 = []
57
58 int_stai1 = []
59 int_stai2 = []
60 int_stai3 = []
61 int_stai4 = []
62 int_stai5 = []
63 int_stai6 = []

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

64
65 total_stai = []
66
67 timestamps = []
68
69 stroop = []
70
71 #set window
72 win = psychopy.visual.Window(size = [1000, 800], units = "pix", fullscr = False, color = [1,
    1, 1])
73
74 #INTRODUCTION WINDOW
75 intro_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
76 intro_txt.text = """
77 Welcome to the Experiment! \n
78 Press space to begin.
79 """
80
81 intro_txt.draw()
82 win.flip()
83
84 curr_time = time.time()
85 timestamps.append(curr_time)
86
87 psychopy.event.waitKeys(keyList=['space'])
88
89 #ACCLIMATION PHASE
90 acclimation_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
91 acclimation_txt.text = """
92 ACCLIMATION PHASE \n
93 This phase will take around 5 minutes to acclimate you. \n
94 All the instructions of the experiment are shown on the screen. \n
95 Please move as little as possible during the experiment.
96 """
97
98 acclimation_txt.draw()
99 win.flip()
100
101 curr_time = time.time()
102 timestamps.append(curr_time)
103
104 psychopy.core.wait(300) #300
105
106 acclimation_txt.text = """
107 Press space to continue.
108 """
109
110 acclimation_txt.draw()
111 win.flip()
112
113 psychopy.event.waitKeys(keyList=['space'])
114
115 #BASELINE MEASUREMENT
116 baseline_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
117 baseline_txt.text = """
118 BASELINE MEASUREMENT \n
119 This phase will take about 5 minutes. \n
120 Please move as little as possible.
121 """
122
123 baseline_txt.draw()
124 win.flip()
125

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

126 curr_time = time.time()
127 timestamps.append(curr_time)
128 print(timestamps)
129
130 psychopy.core.wait(300) #300
131
132 baseline_txt.text = """
133 Press space to continue.
134 """
135
136 baseline_txt.draw()
137 win.flip()
138
139 curr_time = time.time()
140 timestamps.append(curr_time)
141
142 psychopy.event.waitKeys(keyList=['space'])
143
144 #STAI QUESTIONNAIRE
145 stai_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
146 stai_txt.text = """
147 Short State-Trait Anxiety Inventory \n
148 To select one of the answers, press key 1-4. \n
149 Press space to continue.
150 """
151
152 stai_txt.draw()
153 win.flip()
154
155 curr_time = time.time()
156 timestamps.append(curr_time)
157
158 psychopy.event.waitKeys(keyList=['space'])
159
160 #item1
161 stai_item1_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
162 stai_item1_txt.text = """
163 I feel calm ... \n \n
164 1. Not at all \n
165 2. Somewhat \n
166 3. Moderately \n
167 4. Very much \n
168 """
169
170 stai_item1_txt.draw()
171 win.flip()
172
173 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
174 stai_1.append(keys[0])
175 print(keys)
176
177 #item2
178 stai_item2_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
179 stai_item2_txt.text = """
180 I am tense ... \n \n
181 1. Not at all \n
182 2. Somewhat \n
183 3. Moderately \n
184 4. Very much \n
185 """
186
187 stai_item2_txt.draw()
188 win.flip()

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

189
190 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
191 stai_1.append(keys[0])
192 print(keys)
193
194 #item3
195 stai_item3_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
196 stai_item3_txt.text = """
197 I feel upset ... \n \n
198 1. Not at all \n
199 2. Somewhat \n
200 3. Moderately \n
201 4. Very much \n
202 """
203
204 stai_item3_txt.draw()
205 win.flip()
206
207 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
208 stai_1.append(keys[0])
209 print(keys)
210
211 #item4
212 stai_item4_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
213 stai_item4_txt.text = """
214 I am relaxed ... \n \n
215 1. Not at all \n
216 2. Somewhat \n
217 3. Moderately \n
218 4. Very much \n
219 """
220
221 stai_item4_txt.draw()
222 win.flip()
223
224 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
225 stai_1.append(keys[0])
226 print(keys)
227
228 #item5
229 stai_item5_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
230 stai_item5_txt.text = """
231 I feel content ... \n \n
232 1. Not at all \n
233 2. Somewhat \n
234 3. Moderately \n
235 4. Very much \n
236 """
237
238 stai_item5_txt.draw()
239 win.flip()
240
241 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
242 stai_1.append(keys[0])
243 print(keys)
244
245 #item6
246 stai_item6_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
247 stai_item6_txt.text = """
248 I am worried ... \n \n
249 1. Not at all \n
250 2. Somewhat \n
251 3. Moderately \n

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

252 4. Very much \n
253 """
254
255 stail_item6_txt.draw()
256 win.flip()
257
258 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
259 stail_1.append(keys[0])
260 print(keys)
261 print(stail_1)
262
263 #ALLEGED SING-A-SONG STRESS TEST
264 sasst_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
265 sasst_txt.text = """
266 SING-A-SONG STRESS TEST \n
267 For this task you have to first think of a song and then perform 30 seconds of that song. \n
268 Therefore, you have two minutes to think of a song. \n
269 Afterwards, you have to perform 30 seconds of that song. \n
270 Press space to begin.
271 """
272
273 sasst_txt.draw()
274 win.flip()
275
276 curr_time = time.time()
277 timestamps.append(curr_time)
278
279 psychopy.event.waitKeys(keyList=['space'])
280
281 sasst_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
282 sasst_txt.text = """
283 You have two minutes to think of a song. \n
284 This can be any song you like or know.
285 """
286
287 sasst_txt.draw()
288 win.flip()
289
290 curr_time = time.time()
291 timestamps.append(curr_time)
292
293 psychopy.core.wait(120) #120
294
295 sasst_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
296 sasst_txt.text = """
297 Are you ready? \n
298 Press space to sing the song.
299 """
300
301 sasst_txt.draw()
302 win.flip()
303
304 curr_time = time.time()
305 timestamps.append(curr_time)
306
307 psychopy.event.waitKeys(keyList=['space'])
308
309 sasst_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
310 sasst_txt.text = """
311 Good news for you! \n
312 You do not have perform the song. \n
313 Press space to continue.
314 """

```


VALIDATION OF A WRIST-WORN PPG SENSOR

```

315
316 sasst_txt.draw()
317 win.flip()
318
319 curr_time = time.time()
320 timestamps.append(curr_time)
321
322 psychopy.event.waitKeys(keyList=['space'])
323
324 #STAI QUESTIONNAIRE
325 stai_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
326 stai_txt.text = """
327 Short State-Trait Anxiety Inventory \n
328 To select one of the answers, press key 1-4. \n
329 Press space to continue.
330 """
331
332 stai_txt.draw()
333 win.flip()
334
335 curr_time = time.time()
336 timestamps.append(curr_time)
337
338 psychopy.event.waitKeys(keyList=['space'])
339
340 #item1
341 stai_item1_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
342 stai_item1_txt.text = """
343 I feel calm ... \n \n
344 1. Not at all \n
345 2. Somewhat \n
346 3. Moderately \n
347 4. Very much \n
348 """
349
350 stai_item1_txt.draw()
351 win.flip()
352
353 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
354 stai_2.append(keys[0])
355 print(keys)
356
357 #item2
358 stai_item2_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
359 stai_item2_txt.text = """
360 I am tense ... \n \n
361 1. Not at all \n
362 2. Somewhat \n
363 3. Moderately \n
364 4. Very much \n
365 """
366
367 stai_item2_txt.draw()
368 win.flip()
369
370 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
371 stai_2.append(keys[0])
372 print(keys)
373
374 #item3
375 stai_item3_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
376 stai_item3_txt.text = """
377 I feel upset ... \n \n

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

378 1. Not at all \n
379 2. Somewhat \n
380 3. Moderately \n
381 4. Very much \n
382 """
383
384 stai_item3.txt.draw()
385 win.flip()
386
387 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
388 stai_2.append(keys[0])
389 print(keys)
390
391 #item4
392 stai_item4.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
393 stai_item4.txt.text = """
394 I am relaxed ... \n \n
395 1. Not at all \n
396 2. Somewhat \n
397 3. Moderately \n
398 4. Very much \n
399 """
400
401 stai_item4.txt.draw()
402 win.flip()
403
404 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
405 stai_2.append(keys[0])
406 print(keys)
407
408 #item5
409 stai_item5.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
410 stai_item5.txt.text = """
411 I feel content ... \n \n
412 1. Not at all \n
413 2. Somewhat \n
414 3. Moderately \n
415 4. Very much \n
416 """
417
418 stai_item5.txt.draw()
419 win.flip()
420
421 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
422 stai_2.append(keys[0])
423 print(keys)
424
425 #item6
426 stai_item6.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
427 stai_item6.txt.text = """
428 I am worried ... \n \n
429 1. Not at all \n
430 2. Somewhat \n
431 3. Moderately \n
432 4. Very much \n
433 """
434
435 stai_item6.txt.draw()
436 win.flip()
437
438 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
439 stai_2.append(keys[0])
440 print(keys)

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

441 print(stai_2)
442
443 #RECOVERY PHASE
444 recovery_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
445 recovery_txt.text = """
446 RECOVERY PHASE \n
447 This phase will take about 5 minutes.
448 """
449
450 recovery_txt.draw()
451 win.flip()
452
453 curr_time = time.time()
454 timestamps.append(curr_time)
455
456 psychopy.core.wait(300) #300
457
458 recovery_txt.text = """
459 Press space to continue.
460 """
461
462 recovery_txt.draw()
463 win.flip()
464
465 psychopy.event.waitKeys(keyList=['space'])
466
467 #STAI QUESTIONNAIRE
468 stai_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
469 stai_txt.text = """
470 Short State-Trait Anxiety Inventory \n
471 To select one of the answers, press key 1-4. \n
472 Press space to continue.
473 """
474
475 stai_txt.draw()
476 win.flip()
477
478 curr_time = time.time()
479 timestamps.append(curr_time)
480
481 psychopy.event.waitKeys(keyList=['space'])
482
483 #item1
484 stai_item1_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
485 stai_item1_txt.text = """
486 I feel calm ... \n \n
487 1. Not at all \n
488 2. Somewhat \n
489 3. Moderately \n
490 4. Very much \n
491 """
492
493 stai_item1_txt.draw()
494 win.flip()
495
496 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
497 stai_3.append(keys[0])
498 print(keys)
499
500 #item2
501 stai_item2_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
502 stai_item2_txt.text = """
503 I am tense ... \n \n

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

504 1. Not at all \n
505 2. Somewhat \n
506 3. Moderately \n
507 4. Very much \n
508 """
509
510 stai_item2.txt.draw()
511 win.flip()
512
513 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
514 stai_3.append(keys[0])
515 print(keys)
516
517 #item3
518 stai_item3.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
519 stai_item3.txt.text = """
520 I feel upset ... \n \n
521 1. Not at all \n
522 2. Somewhat \n
523 3. Moderately \n
524 4. Very much \n
525 """
526
527 stai_item3.txt.draw()
528 win.flip()
529
530 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
531 stai_3.append(keys[0])
532 print(keys)
533
534 #item4
535 stai_item4.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
536 stai_item4.txt.text = """
537 I am relaxed ... \n \n
538 1. Not at all \n
539 2. Somewhat \n
540 3. Moderately \n
541 4. Very much \n
542 """
543
544 stai_item4.txt.draw()
545 win.flip()
546
547 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
548 stai_3.append(keys[0])
549 print(keys)
550
551 #item5
552 stai_item5.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
553 stai_item5.txt.text = """
554 I feel content ... \n \n
555 1. Not at all \n
556 2. Somewhat \n
557 3. Moderately \n
558 4. Very much \n
559 """
560
561 stai_item5.txt.draw()
562 win.flip()
563
564 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
565 stai_3.append(keys[0])
566 print(keys)

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

567
568 #item6
569 stai_item6_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
570 stai_item6_txt.text = """
571 I am worried ... \n \n
572 1. Not at all \n
573 2. Somewhat \n
574 3. Moderately \n
575 4. Very much \n
576 """
577
578 stai_item6_txt.draw()
579 win.flip()
580
581 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
582 stai_3.append(keys[0])
583 print(keys)
584 print(stai_3)
585
586 #BASELINE MEASUREMENT
587 baseline_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
588 baseline_txt.text = """
589 BASELINE MEASUREMENT \n
590 This phase will take about 5 minutes. \n
591 Please move as little as possible.
592 """
593
594 baseline_txt.draw()
595 win.flip()
596
597 curr_time = time.time()
598 timestamps.append(curr_time)
599
600 psychopy.core.wait(300) #300
601
602 baseline_txt.text = """
603 Press space to continue. \n
604 """
605
606 curr_time = time.time()
607 timestamps.append(curr_time)
608
609 baseline_txt.draw()
610 win.flip()
611
612 psychopy.event.waitKeys(keyList=['space'])
613
614 #STAI QUESTIONNAIRE
615 stai_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
616 stai_txt.text = """
617 Short State-Trait Anxiety Inventory \n
618 To select one of the answers, press key 1-4. \n
619 Press space to continue.
620 """
621
622 stai_txt.draw()
623 win.flip()
624
625 curr_time = time.time()
626 timestamps.append(curr_time)
627
628 psychopy.event.waitKeys(keyList=['space'])
629

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

630 #item1
631 stai_item1.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
632 stai_item1.txt.text = """
633 I feel calm ... \n \n
634 1. Not at all \n
635 2. Somewhat \n
636 3. Moderately \n
637 4. Very much \n
638 """
639
640 stai_item1.txt.draw()
641 win.flip()
642
643 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
644 stai_4.append(keys[0])
645 print(keys)
646
647 #item2
648 stai_item2.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
649 stai_item2.txt.text = """
650 I am tense ... \n \n
651 1. Not at all \n
652 2. Somewhat \n
653 3. Moderately \n
654 4. Very much \n
655 """
656
657 stai_item2.txt.draw()
658 win.flip()
659
660 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
661 stai_4.append(keys[0])
662 print(keys)
663
664 #item3
665 stai_item3.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
666 stai_item3.txt.text = """
667 I feel upset ... \n \n
668 1. Not at all \n
669 2. Somewhat \n
670 3. Moderately \n
671 4. Very much \n
672 """
673
674 stai_item3.txt.draw()
675 win.flip()
676
677 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
678 stai_4.append(keys[0])
679 print(keys)
680
681 #item4
682 stai_item4.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
683 stai_item4.txt.text = """
684 I am relaxed ... \n \n
685 1. Not at all \n
686 2. Somewhat \n
687 3. Moderately \n
688 4. Very much \n
689 """
690
691 stai_item4.txt.draw()
692 win.flip()

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

693
694 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
695 stai_4.append(keys[0])
696 print(keys)
697
698 #item5
699 stai_item5.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
700 stai_item5.txt.text = """
701 I feel content ... \n \n
702 1. Not at all \n
703 2. Somewhat \n
704 3. Moderately \n
705 4. Very much \n
706 """
707
708 stai_item5.txt.draw()
709 win.flip()
710
711 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
712 stai_4.append(keys[0])
713 print(keys)
714
715 #item6
716 stai_item6.txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
717 stai_item6.txt.text = """
718 I am worried ... \n \n
719 1. Not at all \n
720 2. Somewhat \n
721 3. Moderately \n
722 4. Very much \n
723 """
724
725 stai_item6.txt.draw()
726 win.flip()
727
728 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
729 stai_4.append(keys[0])
730 print(keys)
731 print(stai_4)
732
733 #STROOP-COLOR-WORD TEST
734 # Colors abd screen
735 col.white = (250, 250, 250)
736 col.black = (0, 0, 0)
737 col.gray = (220, 220, 220)
738 col.red = (250, 0, 0)
739 col.green = (0, 200, 0)
740 col.blue = (0, 0, 250)
741 col.yellow = (250,250,0)
742
743 BACKGR.COL = col.gray
744 SCREEN.SIZE = (700, 500)
745
746
747 # Experiment
748 n_trials = 500
749
750 WORDS = ("red", "green", "blue")
751
752 COLORS = {"red": col.red,
753           "green": col.green,
754           "blue": col.blue}
755

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

756 KEYS      = {"red": K_b,
757              "green": K_n,
758              "blue": K_m}
759
760 pygame.init()
761 pygame.display.set_mode(SCREEN_SIZE)
762 pygame.display.set_caption("Stroop Test")
763
764 screen = pygame.display.get_surface()
765 # screen.fill(BACKGR_COL)
766
767 font = pygame.font.Font(None, 80)
768 font_small = pygame.font.Font(None, 40)
769
770 curr_time = time.time()
771 timestamps.append(curr_time)
772
773 def main():
774
775     STATE = "welcome"
776     trial_number = 0
777
778     while True:
779
780         # refreshing the surface
781         screen.fill(BACKGR_COL)
782
783         # Event loop
784         for event in pygame.event.get():
785
786             # interactive transitionals
787             if STATE == "welcome":
788                 if event.type == KEYDOWN and event.key == K_SPACE:
789                     STATE = "prepare_trial"
790                     print(STATE)
791                     continue
792
793             if STATE == "trial":
794                 if event.type == KEYDOWN and event.key in KEYS.values():
795                     time_when_reacted = time.time()
796                     this_reaction_time = time_when_reacted - time_when_presented
797                     this_correctness = (event.key == KEYS[this_color])
798                     if trial_number < n_trials:
799                         STATE = "prepare_trial"
800                     else:
801                         STATE = "goodbye"
802                     print(STATE)
803                     continue
804
805             if event.type == QUIT:
806                 STATE = "quit"
807                 print(STATE)
808                 break
809
810
811         # automatic transitionals
812         if STATE == "prepare_trial":
813             trial_number = trial_number + 1
814             this_word = pick_color()
815             this_color = pick_color()
816             time_when_presented = time.time()
817             STATE = "trial"
818

```


VALIDATION OF A WRIST-WORN PPG SENSOR

```

819         print (STATE)
820
821
822
823     # Presentitionals
824     if STATE == "welcome":
825         draw_welcome()
826         draw_button(SCREEN_SIZE[0]*1/5, 450, "Red: B", col_red)
827         draw_button(SCREEN_SIZE[0]*3/5, 450, "Green: N", col_green)
828         draw_button(SCREEN_SIZE[0]*4/5, 450, "Blue: M", col_blue)
829
830     if STATE == "trial":
831         draw_stimulus(this_color, this_word)
832         draw_button(SCREEN_SIZE[0]*1/5, 450, "Red: B", col_red)
833         draw_button(SCREEN_SIZE[0]*3/5, 450, "Green: N", col_green)
834         draw_button(SCREEN_SIZE[0]*4/5, 450, "Blue: M", col_blue)
835
836     if STATE == "goodbye":
837         draw_goodbye()
838
839     if STATE == "quit":
840         pygame.quit()
841         break
842
843     # Updating the display
844     pygame.display.update()
845
846
847 # Function definitions
848
849 def pick_color():
850     """ Return a random word.
851     """
852     random_number = random.randint(0,2)
853     return WORDS[random_number]
854
855 def draw_button(xpos, ypos, label, color):
856     text = font_small.render(label, True, color, BACKGR_COL)
857     text_rectangle = text.get_rect()
858     text_rectangle.center = (xpos, ypos)
859     screen.blit(text, text_rectangle)
860
861 def draw_welcome():
862     text_surface = font.render("STROOP Experiment", True, col_black, BACKGR_COL)
863     text_rectangle = text_surface.get_rect()
864     text_rectangle.center = (SCREEN_SIZE[0]/2.0,150)
865     screen.blit(text_surface, text_rectangle)
866     text_surface = font_small.render("Press Spacebar to continue", True, col_black,
867     BACKGR_COL)
868     text_rectangle = text_surface.get_rect()
869     text_rectangle.center = (SCREEN_SIZE[0]/2.0,300)
870     screen.blit(text_surface, text_rectangle)
871
872 def draw_stimulus(color, word):
873     text_surface = font.render(word, True, COLORS[color], col_gray)
874     text_rectangle = text_surface.get_rect()
875     text_rectangle.center = (SCREEN_SIZE[0]/2.0,150)
876     screen.blit(text_surface, text_rectangle)
877
878 def draw_goodbye():
879     text_surface = font_small.render("END OF THE EXPERIMENT", True, col_black, BACKGR_COL)
880     text_rectangle = text_surface.get_rect()
881     text_rectangle.center = (SCREEN_SIZE[0]/2.0,150)

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

881     screen.blit(text_surface, text_rectangle)
882     text_surface = font_small.render("Close the application.", True, col_black, BACKGR.COL)
883     text_rectangle = text_surface.get_rect()
884     text_rectangle.center = (SCREEN_SIZE[0]/2.0, 200)
885     screen.blit(text_surface, text_rectangle)
886
887 main()
888
889 curr_time = time.time()
890 timestamps.append(curr_time)
891
892 #STAI QUESTIONNAIRE
893 stai_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
894 stai_txt.text = """
895 Short State-Trait Anxiety Inventory \n
896 To select one of the answers, press key 1-4. \n
897 Press space to continue.
898 """
899
900 stai_txt.draw()
901 win.flip()
902
903 curr_time = time.time()
904 timestamps.append(curr_time)
905
906 psychopy.event.waitKeys(keyList=['space'])
907
908 #item1
909 stai_item1_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
910 stai_item1_txt.text = """
911 I feel calm ... \n \n
912 1. Not at all \n
913 2. Somewhat \n
914 3. Moderately \n
915 4. Very much \n
916 """
917
918 stai_item1_txt.draw()
919 win.flip()
920
921 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
922 stai_5.append(keys[0])
923 print(keys)
924
925 #item2
926 stai_item2_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
927 stai_item2_txt.text = """
928 I am tense ... \n \n
929 1. Not at all \n
930 2. Somewhat \n
931 3. Moderately \n
932 4. Very much \n
933 """
934
935 stai_item2_txt.draw()
936 win.flip()
937
938 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
939 stai_5.append(keys[0])
940 print(keys)
941
942 #item3
943 stai_item3_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

944 stai_item3_txt.text = """
945 I feel upset ... \n \n
946 1. Not at all \n
947 2. Somewhat \n
948 3. Moderately \n
949 4. Very much \n
950 """
951
952 stai_item3_txt.draw()
953 win.flip()
954
955 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
956 stai_5.append(keys[0])
957 print(keys)
958
959 #item4
960 stai_item4_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
961 stai_item4_txt.text = """
962 I am relaxed ... \n \n
963 1. Not at all \n
964 2. Somewhat \n
965 3. Moderately \n
966 4. Very much \n
967 """
968
969 stai_item4_txt.draw()
970 win.flip()
971
972 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
973 stai_5.append(keys[0])
974 print(keys)
975
976 #item5
977 stai_item5_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
978 stai_item5_txt.text = """
979 I feel content ... \n \n
980 1. Not at all \n
981 2. Somewhat \n
982 3. Moderately \n
983 4. Very much \n
984 """
985
986 stai_item5_txt.draw()
987 win.flip()
988
989 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
990 stai_5.append(keys[0])
991 print(keys)
992
993 #item6
994 stai_item6_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
995 stai_item6_txt.text = """
996 I am worried ... \n \n
997 1. Not at all \n
998 2. Somewhat \n
999 3. Moderately \n
1000 4. Very much \n
1001 """
1002
1003 stai_item6_txt.draw()
1004 win.flip()
1005
1006 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1007 stai_5.append(keys[0])
1008 print(keys)
1009 print(stai_5)
1010
1011 #RECOVERY PHASE
1012 recovery_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
1013 recovery_txt.text = """
1014 RECOVERY PHASE \n
1015 This phase will take about 5 minutes.
1016 """
1017
1018 recovery_txt.draw()
1019 win.flip()
1020
1021 curr_time = time.time()
1022 timestamps.append(curr_time)
1023
1024 psychopy.core.wait(300) #300
1025
1026 recovery_txt.text = """
1027 Press space to continue.
1028 """
1029
1030 recovery_txt.draw()
1031 win.flip()
1032
1033 psychopy.event.waitKeys(keyList=['space'])
1034
1035 #STAI QUESTIONNAIRE
1036 stai_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
1037 stai_txt.text = """
1038 Short State-Trait Anxiety Inventory \n
1039 To select one of the answers, press key 1-4. \n
1040 Press space to continue.
1041 """
1042
1043 stai_txt.draw()
1044 win.flip()
1045
1046 curr_time = time.time()
1047 timestamps.append(curr_time)
1048
1049 psychopy.event.waitKeys(keyList=['space'])
1050
1051 #item1
1052 stai_item1_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
1053 stai_item1_txt.text = """
1054 I feel calm ... \n \n
1055 1. Not at all \n
1056 2. Somewhat \n
1057 3. Moderately \n
1058 4. Very much \n
1059 """
1060
1061 stai_item1_txt.draw()
1062 win.flip()
1063
1064 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
1065 stai_6.append(keys[0])
1066 print(keys)
1067
1068 #item2
1069 stai_item2_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1070 stai_item2_txt.text = """
1071 I am tense ... \n \n
1072 1. Not at all \n
1073 2. Somewhat \n
1074 3. Moderately \n
1075 4. Very much \n
1076 """
1077
1078 stai_item2_txt.draw()
1079 win.flip()
1080
1081 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
1082 stai_6.append(keys[0])
1083 print(keys)
1084
1085 #item3
1086 stai_item3_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
1087 stai_item3_txt.text = """
1088 I feel upset ... \n \n
1089 1. Not at all \n
1090 2. Somewhat \n
1091 3. Moderately \n
1092 4. Very much \n
1093 """
1094
1095 stai_item3_txt.draw()
1096 win.flip()
1097
1098 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
1099 stai_6.append(keys[0])
1100 print(keys)
1101
1102 #item4
1103 stai_item4_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
1104 stai_item4_txt.text = """
1105 I am relaxed ... \n \n
1106 1. Not at all \n
1107 2. Somewhat \n
1108 3. Moderately \n
1109 4. Very much \n
1110 """
1111
1112 stai_item4_txt.draw()
1113 win.flip()
1114
1115 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
1116 stai_6.append(keys[0])
1117 print(keys)
1118
1119 #item5
1120 stai_item5_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
1121 stai_item5_txt.text = """
1122 I feel content ... \n \n
1123 1. Not at all \n
1124 2. Somewhat \n
1125 3. Moderately \n
1126 4. Very much \n
1127 """
1128
1129 stai_item5_txt.draw()
1130 win.flip()
1131
1132 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1133 stai_6.append(keys[0])
1134 print(keys)
1135
1136 #item6
1137 stai_item6_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
1138 stai_item6_txt.text = """
1139 I am worried ... \n \n
1140 1. Not at all \n
1141 2. Somewhat \n
1142 3. Moderately \n
1143 4. Very much \n
1144 """
1145
1146 stai_item6_txt.draw()
1147 win.flip()
1148
1149 keys = psychopy.event.waitKeys(keyList=["1", "2", "3", "4"])
1150 stai_6.append(keys[0])
1151 print(keys)
1152 print(stai_6)
1153
1154 curr_time = time.time()
1155 timestamps.append(curr_time)
1156
1157 #CLOSING WINDOW
1158 end_txt = psychopy.visual.TextStim(win = win, wrapWidth = 350, color = [-1, -1, -1])
1159 end_txt.text = """
1160 Thank you for participating! \n
1161 You reached the end of the Experiment. \n
1162 Press space to close the application.
1163 """
1164
1165 end_txt.draw()
1166 win.flip()
1167
1168 curr_time = time.time()
1169 timestamps.append(curr_time)
1170
1171 psychopy.event.waitKeys(keyList=['space'])
1172
1173 #save data to csv-files
1174 np.savetxt("p_" + subj_id + "_time_data.csv", timestamps, delimiter=",")
1175
1176 for x in stai_1:
1177     a = int(x)
1178     int_stai1.append(a)
1179
1180 for x in stai_2:
1181     a = int(x)
1182     int_stai2.append(a)
1183
1184 for x in stai_3:
1185     a = int(x)
1186     int_stai3.append(a)
1187
1188 for x in stai_4:
1189     a = int(x)
1190     int_stai4.append(a)
1191
1192 for x in stai_5:
1193     a = int(x)
1194     int_stai5.append(a)
1195

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```
1196 for x in stai_6:
1197     a = int(x)
1198     int_stai6.append(a)
1199
1200 np.savetxt("p_" + subj_id + "_stai1_data.csv", int_stai1, delimiter=",")
1201 np.savetxt("p_" + subj_id + "_stai2_data.csv", int_stai2, delimiter=",")
1202 np.savetxt("p_" + subj_id + "_stai3_data.csv", int_stai3, delimiter=",")
1203 np.savetxt("p_" + subj_id + "_stai4_data.csv", int_stai4, delimiter=",")
1204 np.savetxt("p_" + subj_id + "_stai5_data.csv", int_stai5, delimiter=",")
1205 np.savetxt("p_" + subj_id + "_stai6_data.csv", int_stai6, delimiter=",")
1206
1207 win.close()
```

VALIDATION OF A WRIST-WORN PPG SENSOR

Appendix C – Normalized STAI-Scores Graphic (R-Script)

```

1 rm(list = ls())
2 #####Task1
3 library(ggplot2)
4 library(grDevices)
5 install.packages("gdata")
6 library(gdata)
7 #perl <- gdata:::findPerl("perl")
8 stai <- read.xls("STAI_Normalized.xls", sheet=1)
9
10 quartz(width=10, height=5)
11 ggplot(stai, aes(x = factor(Phase, level = c('T1 (before SSST)', 'T2 (after SSST)', 'T3 (
    after Recovery)',
12                                     'T4 (before Stroop)', 'T5 (after Stroop)', 'T6
    (after Recovery)')), group=1)) +
13   geom_line(aes(y=p1), size = 1, color= "darkgrey") +
14   geom_line(aes(y=p2), size = 1, color= "darkgrey")+
15   geom_line(aes(y=p3), size = 1, color= "darkgrey")+
16   geom_line(aes(y=p4), size = 1, color= "darkgrey")+
17   geom_line(aes(y=p5), size = 1, color= "darkgrey")+
18   geom_line(aes(y=p6), size = 1, color= "darkgrey")+
19   geom_line(aes(y=p7), size = 1, color= "darkgrey")+
20   geom_line(aes(y=p8), size = 1, color= "darkgrey")+
21   geom_line(aes(y=p9), size = 1, color= "darkgrey")+
22   geom_line(aes(y=p10), size = 1, color= "darkgrey")+
23   geom_line(aes(y=p11), size = 1, color= "darkgrey")+
24   geom_line(aes(y=p12), size = 1, color= "darkgrey")+
25   geom_line(aes(y=p13), size = 1, color= "darkgrey")+
26   geom_line(aes(y=p14), size = 1, color= "darkgrey")+
27   geom_line(aes(y=p15), size = 1, color= "darkgrey")+
28   geom_line(aes(y=p16), size = 1, color= "darkgrey")+
29   geom_line(aes(y=p17), size = 1, color= "darkgrey")+
30   geom_line(aes(y=p18), size = 1, color= "darkgrey")+
31   geom_line(aes(y=p19), size = 1, color= "darkgrey")+
32   geom_line(aes(y=p20), size = 1, color= "darkgrey")+
33   geom_line(aes(y=mean), size= 1, color = "red")+
34   labs(title = "STAI", x="", y = "Anxiety")+
35   theme(axis.text.x = element_text(angle = 70, hjust = 1))

```


VALIDATION OF A WRIST-WORN PPG SENSOR

Appendix D – Visual Inspection Graphs (R-Script)

```

1 rm(list = ls())
2 #####PPG_BS1#####
3
4 path <- "Masterarbeit/MASTERDATA/PPG/csv/"
5 files_e4 <- c(list.files(path)[grepl('PPG_E4_BS1', list.files(path))])
6 files_s3 <- c(list.files(path)[grepl('PPG_S3_BS1', list.files(path))])
7 name_e4 <- gsub("PPG_E4_BS1", "", gsub(".csv", "", files_e4))
8 name_s3 <- gsub("PPG_S3_BS1", "", gsub(".csv", "", files_s3))
9 for (u in 1:length(files_e4)){
10   e4_bs1 <-read.csv(paste0(path, files_e4[u]), header=TRUE, sep = ",", quote = "\"", dec =
      ".", fill = TRUE)
11
12 for (o in 1:length(files_s3)){
13   s3_bs1 <-read.csv(paste0(path, files_s3[o]), header=TRUE, sep = ",", quote = "\"", dec =
      ".", fill = TRUE)
14
15   ###Zeit dern
16   e4_bs1$time <- as.POSIXct(as.numeric(e4_bs1$TIME)/1000, origin = '1970-01-01', tz = 'GMT')
17   s3_bs1$time <- as.POSIXct(as.numeric(s3_bs1$TIME)/1000, origin = '1970-01-01', tz = 'GMT')
18
19   ###plots erstellen
20   jpeg(filename=paste0("Masterarbeit/MASTERDATA/PPG/Plots/BS1/", "plot_ppg_bs1_s3", name_s3[o]
      ), ".jpeg"), width = 1000, height = 500)
21   plot(s3_bs1$time, s3_bs1$PPG_S3_BS1, type = "l", main = paste0("Plot PPG S3 BS1", name_s3[o]
      ), xlab = "zeit", ylab = "s3_bs1")
22   dev.off()
23
24   jpeg(filename=paste0("Masterarbeit/MASTERDATA/PPG/Plots/BS1/", "plot_ppg_bs1_e4", name_e4[u]
      ), ".jpeg"), width = 1000, height = 500)
25   plot(e4_bs1$time, e4_bs1$PPG_E4_BS1, type = "l", main = paste0("Plot PPG E4 BS1", name_e4[u]
      ), xlab = "zeit", ylab = "e4_bs1")
26   dev.off()
27
28 save(file=paste0("Masterarbeit/MASTERDATA/PPG/S3/S3_BS1", name_s3[o], ".RData"), s3_bs1,
      compress = T)
29 }#ende for files_s3
30
31 save(file=paste0("Masterarbeit/MASTERDATA/PPG/E4/E4_BS1", name_e4[u], ".RData"), e4_bs1,
      compress = T)
32 }#ende for files_e4
33
34
35 rm(list = ls())
36 #####PPG_ST1#####
37
38 path <- "Masterarbeit/MASTERDATA/PPG/csv/"
39 files_e4 <- c(list.files(path)[grepl('PPG_E4_ST1', list.files(path))])
40 files_s3 <- c(list.files(path)[grepl('PPG_S3_ST1', list.files(path))])
41 name_e4 <- gsub("PPG_E4_ST1", "", gsub(".csv", "", files_e4))
42 name_s3 <- gsub("PPG_S3_ST1", "", gsub(".csv", "", files_s3))
43 for (u in 1:length(files_e4)){
44   e4_st1 <-read.csv(paste0(path, files_e4[u]), header=TRUE, sep = ",", quote = "\"", dec =
      ".", fill = TRUE)
45
46   for (o in 1:length(files_s3)){
47     s3_st1 <-read.csv(paste0(path, files_s3[o]), header=TRUE, sep = ",", quote = "\"", dec =
      ".", fill = TRUE)
48
49     ###Zeit dern
50     e4_st1$time <- as.POSIXct(as.numeric(e4_st1$TIME)/1000, origin = '1970-01-01', tz = 'GMT'
      ')
51     s3_st1$time <- as.POSIXct(as.numeric(s3_st1$TIME)/1000, origin = '1970-01-01', tz = 'GMT'
      ')

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

52
53   ###plots erstellen
54   jpeg(filename=paste0(" Masterarbeit/MASTERDATA/PPG/Plots/ST1/", "plot_ppg-st1-s3", name_s3
55   [o], ".jpeg"), width = 1000, height = 500)
56   plot(s3_st1$time, s3_st1$PPG_S3_ST1, type = "l", main = paste0("Plot PPG S3 ST1", name_s3[
57   o]), xlab = "zeit", ylab = "s3_st1")
58   dev.off()
59
60   jpeg(filename=paste0(" Masterarbeit/MASTERDATA/PPG/Plots/ST1/", "plot_ppg-st1-e4", name_e4
61   [u], ".jpeg"), width = 1000, height = 500)
62   plot(e4_st1$time, e4_st1$PPG_E4_ST1, type = "l", main = paste0("Plot PPG E4 ST1", name_e4
63   [u]), xlab = "zeit", ylab = "e4_st1")
64   dev.off()
65
66   save(file=paste0(" Masterarbeit/MASTERDATA/PPG/S3/S3_ST1", name_s3[o], ".RData"), s3_st1,
67   compress = T)
68 }#ende for files_s3
69 save(file=paste0(" Masterarbeit/MASTERDATA/PPG/E4/E4_ST1", name_e4[u], ".RData"), e4_st1,
70 compress = T)
71 }#ende for files_e4
72
73 rm(list = ls())
74 #####PPG_RC1#####
75 path <- " Masterarbeit/MASTERDATA/PPG/csv/"
76 files_e4 <- c(list.files(path)[grepl('PPG_E4_RC1', list.files(path))])
77 files_s3 <- c(list.files(path)[grepl('PPG_S3_RC1', list.files(path))])
78 name_e4 <- gsub("PPG_E4_RC1", "", gsub(".csv", "", files_e4))
79 name_s3 <- gsub("PPG_S3_RC1", "", gsub(".csv", "", files_s3))
80 for (u in 1:length(files_e4)){
81   e4_rc1 <- read.csv(paste0(path, files_e4[u]), header=TRUE, sep = ",", quote = "\"", dec =
82   ".", fill = TRUE)
83
84   for (o in 1:length(files_s3)){
85     s3_rc1 <- read.csv(paste0(path, files_s3[o]), header=TRUE, sep = ",", quote = "\"", dec =
86     ".", fill = TRUE)
87
88     ###Zeit dern
89     e4_rc1$time <- as.POSIXct(as.numeric(e4_rc1$TIME)/1000, origin = '1970-01-01', tz = 'GMT
90     ')
91     s3_rc1$time <- as.POSIXct(as.numeric(s3_rc1$TIME)/1000, origin = '1970-01-01', tz = 'GMT
92     ')
93
94     ###plots erstellen
95     jpeg(filename=paste0(" Masterarbeit/MASTERDATA/PPG/Plots/RC1/", "plot_ppg-rc1-s3", name_s3
96     [o], ".jpeg"), width = 1000, height = 500)
97     plot(s3_rc1$time, s3_rc1$PPG_S3_RC1, type = "l", main = paste0("Plot PPG S3 RC1", name_s3[
98     o]), xlab = "zeit", ylab = "s3_rc1")
99     dev.off()
100
101     jpeg(filename=paste0(" Masterarbeit/MASTERDATA/PPG/Plots/RC1/", "plot_ppg-rc1-e4", name_e4
102     [u], ".jpeg"), width = 1000, height = 500)
103     plot(e4_rc1$time, e4_rc1$PPG_E4_RC1, type = "l", main = paste0("Plot PPG E4 RC1", name_e4
104     [u]), xlab = "zeit", ylab = "e4_rc1")
105     dev.off()
106
107     save(file=paste0(" Masterarbeit/MASTERDATA/PPG/S3/S3_RC1", name_s3[o], ".RData"), s3_rc1,
108     compress = T)
109 }#ende for files_s3
110 save(file=paste0(" Masterarbeit/MASTERDATA/PPG/E4/E4_RC1", name_e4[u], ".RData"), e4_rc1,
111 compress = T)
112 }#ende for files_e4
113
114

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

99
100 rm(list = ls())
101 #####PPG_BS2#####
102 path <- "Masterarbeit/MASTERDATA/PPG/csv/"
103 files_e4 <- c(list.files(path)[grepl('PPG_E4_BS2', list.files(path))])
104 files_s3 <- c(list.files(path)[grepl('PPG_S3_BS2', list.files(path))])
105 name_e4 <- gsub("PPG_E4_BS2", "", gsub(".csv", "", files_e4))
106 name_s3 <- gsub("PPG_S3_BS2", "", gsub(".csv", "", files_s3))
107 for (u in 1:length(files_e4)){
108   e4_bs2 <- read.csv(paste0(path, files_e4[u]), header=TRUE, sep = ",", quote = "\"", dec =
    ".", fill = TRUE)
109
110   for (o in 1:length(files_s3)){
111     s3_bs2 <- read.csv(paste0(path, files_s3[o]), header=TRUE, sep = ",", quote = "\"", dec =
    ".", fill = TRUE)
112
113     ###Zeit dern
114     e4_bs2$time <- as.POSIXct(as.numeric(e4_bs2$TIME)/1000, origin = '1970-01-01', tz = 'GMT
    ')
115     s3_bs2$time <- as.POSIXct(as.numeric(s3_bs2$TIME)/1000, origin = '1970-01-01', tz = 'GMT
    ')
116
117
118     ###plots erstellen
119     jpeg(filename=paste0("Masterarbeit/MASTERDATA/PPG/Plots/BS2/", "plot_ppg_bs2_s3", name_s3
    [o], ".jpeg"), width = 1000, height = 500)
120     plot(s3_bs2$time, s3_bs2$PPG_S3_BS2, type = "l", main = paste0("Plot PPG S3 BS2", name_s3[
    o]), xlab = "zeit", ylab = "s3_bs2")
121     dev.off()
122
123     jpeg(filename=paste0("Masterarbeit/MASTERDATA/PPG/Plots/BS2/", "plot_ppg_bs2_e4", name_e4
    [u], ".jpeg"), width = 1000, height = 500)
124     plot(e4_bs2$time, e4_bs2$PPG_E4_BS2, type = "l", main = paste0("Plot PPG E4 BS2", name_e4
    [u]), xlab = "zeit", ylab = "e4_bs2")
125     dev.off()
126
127     save(file=paste0("Masterarbeit/MASTERDATA/PPG/S3/S3_BS2", name_s3[o], ".RData"), s3_bs2,
    compress = T)
128   }#ende for files_s3
129   save(file=paste0("Masterarbeit/MASTERDATA/PPG/E4/E4_BS2", name_e4[u], ".RData"), e4_bs2,
    compress = T)
130 }#ende for files_e4
131
132
133 rm(list = ls())
134 #####PPG_RC#####
135 path <- "Masterarbeit/MASTERDATA/PPG/csv/"
136 files_e4 <- c(list.files(path)[grepl('PPG_E4_ST2', list.files(path))])
137 files_s3 <- c(list.files(path)[grepl('PPG_S3_ST2', list.files(path))])
138 name_e4 <- gsub("PPG_E4_ST2", "", gsub(".csv", "", files_e4))
139 name_s3 <- gsub("PPG_S3_ST2", "", gsub(".csv", "", files_s3))
140 for (u in 1:length(files_e4)){
141   e4_st2 <- read.csv(paste0(path, files_e4[u]), header=TRUE, sep = ",", quote = "\"", dec =
    ".", fill = TRUE)
142
143   for (o in 1:length(files_s3)){
144     s3_st2 <- read.csv(paste0(path, files_s3[o]), header=TRUE, sep = ",", quote = "\"", dec =
    ".", fill = TRUE)
145
146     ###Zeit dern
147     e4_st2$time <- as.POSIXct(as.numeric(e4_st2$TIME)/1000, origin = '1970-01-01', tz = 'GMT
    ')
148     s3_st2$time <- as.POSIXct(as.numeric(s3_st2$TIME)/1000, origin = '1970-01-01', tz = 'GMT

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

    ')
149
150 ###plots erstellen
151 jpeg(filename=paste0(" Masterarbeit/MASTERDATA/PPG/Plots/ST2/", "plot_ppg-st2_s3", name_s3
152 [o], ".jpeg"), width = 1000, height = 500)
153 plot(s3_st2$time, s3_st2$PPG_S3_ST2, type = "l", main = paste0("Plot PPG S3 ST2", name_s3[
154 o]), xlab = "zeit", ylab = "s3_st2")
155 dev.off()
156
157 jpeg(filename=paste0(" Masterarbeit/MASTERDATA/PPG/Plots/ST2/", "plot_ppg-st2_e4", name_e4
158 [u], ".jpeg"), width = 1000, height = 500)
159 plot(e4_st2$time, e4_st2$PPG_E4_ST2, type = "l", main = paste0("Plot PPG E4 ST2", name_e4
160 [u]), xlab = "zeit", ylab = "e4_st2")
161 dev.off()
162
163 save(file=paste0(" Masterarbeit/MASTERDATA/PPG/S3/S3_ST2", name_s3[o], ".RData"), s3_st2,
164 compress = T)
165 }#ende for files_s3
166 save(file=paste0(" Masterarbeit/MASTERDATA/PPG/E4/E4_ST2", name_e4[u], ".RData"), e4_st2,
167 compress = T)
168 }#ende for files_e4
169
170 rm(list = ls())
171 #####PPG_RC2#####
172 path <- " Masterarbeit/MASTERDATA/PPG/csv/"
173 files_e4 <- c(list.files(path)[grepl("PPG_E4_RC2", list.files(path))])
174 files_s3 <- c(list.files(path)[grepl("PPG_S3_RC2", list.files(path))])
175 name_e4 <- gsub("PPG_E4_RC2", "", gsub(".csv", "", files_e4))
176 name_s3 <- gsub("PPG_S3_RC2", "", gsub(".csv", "", files_s3))
177 for (u in 1:length(files_e4)){
178   e4_rc2 <- read.csv(paste0(path, files_e4[u]), header=TRUE, sep = ",", quote = "\"", dec =
179   ".", fill = TRUE)
180
181   for (o in 1:length(files_s3)){
182     s3_rc2 <- read.csv(paste0(path, files_s3[o]), header=TRUE, sep = ",", quote = "\"", dec =
183     ".", fill = TRUE)
184
185     ###Zeit dern
186     e4_rc2$time <- as.POSIXct(as.numeric(e4_rc2$TIME)/1000, origin = '1970-01-01', tz = 'GMT
187     ')
188     s3_rc2$time <- as.POSIXct(as.numeric(s3_rc2$TIME)/1000, origin = '1970-01-01', tz = 'GMT
189     ')
190
191     ###plots erstellen
192     jpeg(filename=paste0(" Masterarbeit/MASTERDATA/PPG/Plots/RC2/", "plot_ppg-rc2_s3", name_s3
193     [o], ".jpeg"), width = 1000, height = 500)
194     plot(s3_rc2$time, s3_rc2$PPG_S3_RC2, type = "l", main = paste0("Plot PPG S3 RC2", name_s3[
195     o]), xlab = "zeit", ylab = "s3_rc2")
196     dev.off()
197
198     jpeg(filename=paste0(" Masterarbeit/MASTERDATA/PPG/Plots/RC2/", "plot_ppg-rc2_e4", name_e4
199     [u], ".jpeg"), width = 1000, height = 500)
200     plot(e4_rc2$time, e4_rc2$PPG_E4_RC2, type = "l", main = paste0("Plot PPG E4 RC2", name_e4
201     [u]), xlab = "zeit", ylab = "e4_rc2")
202     dev.off()
203
204     save(file=paste0(" Masterarbeit/MASTERDATA/PPG/S3/S3_RC2", name_s3[o], ".RData"), s3_rc2,
205     compress = T)
206   }#ende for files_s3
207   save(file=paste0(" Masterarbeit/MASTERDATA/PPG/E4/E4_RC2", name_e4[u], ".RData"), e4_rc2,
208     compress = T)

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```
195 }##ende for files_e4
```

VALIDATION OF A WRIST-WORN PPG SENSOR

Appendix E – Pre-processing & Extraction of Parameters (Python-Script)

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Created on Fri May 8 21:48:01 2020
5
6 @author: Niklas Enewoldsen
7 """
8
9 #IMPORTS
10 import os
11 import csv
12 import time
13 import datetime
14 import os.path
15
16 import numpy as np
17 import scipy as sp
18 import heartpy as hp
19 import matplotlib.pyplot as plt
20
21
22 #SET DIRECTORY
23 os.chdir('/Users/ene/Documents/Master')
24
25 #VARIABLES
26 time_s3_adj = []
27 ppg_s3_adj = []
28 ibi_s3_adj = []
29 hr_s3_adj = []
30
31 bvp_e4_adj = []
32 ibi_e4_adj = []
33 ibt_e4_adj = []
34 hr_e4_adj = []
35
36 time_units = []
37
38 timestamps_bvp = [0,
39                   1558607821000, 1558611680000, 1558615230000, 1558618648000,
40                   1558625271000, 1558687078000, 1558690199000, 1558693158000,
41                   1558697905000, 1558700019000, 1558703299000, 1558706377000,
42                   1558898720000, 1558901276000, 1558906390000, 1558949721000,
43                   1558954011000, 1558956966000, 1558959780000]
44 timestamps_ibi = [0,
45                   1558607821000, 1558611680000, 1558615230000, 1558618648000,
46                   1558625271000, 1558687078000, 1558690199000, 1558693158000,
47                   1558697905000, 1558700019000, 1558703299000, 1558706377000,
48                   1558898720000, 1558901276000, 1558906390000, 1558949721000,
49                   1558954011000, 1558956966000, 1558959780000]
50 timestamps_hr = [0,
51                  1558607831000, 1558611690000, 1558615240000, 1558618658000,
52                  1558625281000, 1558687088000, 1558690209000, 1558693168000,
53                  1558697915000, 1558700029000, 1558703309000, 1558706387000,
54                  1558898730000, 1558901286000, 1558906400000, 1558949731000,
55                  1558954021000, 1558956976000, 1558959790000]
56
57 start_time_bs1 = [0,
58                  1558608133920, 1558611962237, 1558615527392, 1558618965540,
59                  1558625002762, 1558687412138, 1558690509015, 1558693324066,
60                  1558697942775, 1558700305016, 1558703615416, 1558706670308,
61                  1558899026943, 1558901591781, 1558906663802, 1558949990710,
62                  1558954321215, 1558956940916, 1558959756152] #03
63 start_time_st1 = [0,

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

64      1558608473183, 1558612349524, 1558615867258, 1558619321992,
65      1558625333711, 1558687760303, 1558690893903, 1558693659648,
66      1558698285324, 1558700634014, 1558704134017, 1558707003157,
67      1558899399654, 1558901936828, 1558907060205, 1558950373509,
68      1558954672545, 1558957301832, 1558960083546] #06
69 start_time_rc1 = [0,
70      1558608625747, 1558612566845, 1558616084645, 1558619497810,
71      1558625495543, 1558687925302, 1558691087404, 1558693839182,
72      1558698464908, 1558700804581, 1558704299796, 1558707218312,
73      1558899567267, 1558902138162, 1558907259441, 1558950578574,
74      1558954849976, 1558957488180, 1558960248177] #11
75 start_time_bs2 = [0,
76      1558608942642, 1558612884208, 1558616411459, 1558619824108,
77      1558625808373, 1558688250633, 1558691418268, 1558694169314,
78      1558698784139, 1558701132546, 1558704629527, 1558707537008,
79      1558899897594, 1558902472038, 1558907628608, 1558950917919,
80      1558955175853, 1558957822890, 1558960652160] #13
81 start_time_st2 = [0,
82      1558609271761, 1558613196952, 1558616746852, 1558620149699,
83      1558626120623, 1558688587293, 1558691754580, 1558694584715,
84      1558699099464, 1558701455166, 1558705001790, 1558707867963,
85      1558900217589, 1558902830863, 1558907964789, 1558951247997,
86      1558955490820, 1558958165820, 1558960965563] #16
87 start_time_rc2 = [0,
88      1558609629065, 1558613625461, 1558617265974, 1558620663565,
89      1558626439250, 1558689097064, 1558692195459, 1558695001310,
90      1558699535662, 1558701822095, 1558705536296, 1558708560058,
91      1558900612244, 1558903341259, 1558908414307, 1558951746772,
92      1558955914398, 1558958743752, 1558961349168] #19
93
94 end_time_bs1 = [0,
95      1558608434293, 1558612262266, 1558615827422, 1558619265572,
96      1558625302793, 1558687712166, 1558690809044, 1558693624133,
97      1558698242804, 1558700605074, 1558703915446, 1558706970343,
98      1558899326966, 1558901891807, 1558906963830, 1558950290737,
99      1558954621241, 1558957240943, 1558960056197] #04
100 end_time_st1 = [0,
101      1558608602094, 1558612542393, 1558616050208, 1558619463956,
102      1558625472374, 1558687897983, 1558691052200, 1558693805178,
103      1558698438988, 1558700783395, 1558704275144, 1558707198709,
104      1558899540882, 1558902077407, 1558907220936, 1558950544456,
105      1558954818590, 1558957449209, 1558960216890] #10
106 end_time_rc1 = [0,
107      1558608927557, 1558612868456, 1558616387540, 1558619802955,
108      1558625798038, 1558688232165, 1558691388498, 1558694140911,
109      1558698766553, 1558701106726, 1558704611859, 1558707521657,
110      1558899872423, 1558902441451, 1558907585255, 1558950880566,
111      1558955152484, 1558957792405, 1558960633344] #12
112 end_time_bs2 = [0,
113      1558609242679, 1558613184224, 1558616711475, 1558620124124,
114      1558626108389, 1558688550648, 1558691718284, 1558694469363,
115      1558699084154, 1558701432561, 1558704929543, 1558707837037,
116      1558900197613, 1558902772055, 1558907928647, 1558951217943,
117      1558955475869, 1558958122907, 1558960952177] #14
118 end_time_st2 = [0,
119      1558609606225, 1558613596766, 1558617238830, 1558620646297,
120      1558626419967, 1558689070121, 1558692166783, 1558694981943,
121      1558699516354, 1558701805601, 1558705516774, 1558708534361,
122      1558900588289, 1558903314134, 1558908388093, 1558951718944,
123      1558955883535, 1558958699976, 1558961327858] #17
124 end_time_rc2 = [0,
125      1558610040841, 1558613944259, 1558617571186, 1558620965693,
126      1558626743562, 1558689407144, 1558692497904, 1558695291144,

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

127      1558699838774, 1558702123290, 1558705842825, 1558708861870,
128      1558900914170, 1558903649906, 1558908756459, 1558952049213,
129      1558956218257, 1558959049410, 1558961650660] #20
130
131 ppg_s3_bs1 = []
132 ppg_s3_st1 = []
133 ppg_s3_rc1 = []
134 ppg_s3_bs2 = []
135 ppg_s3_st2 = []
136 ppg_s3_rc2 = []
137
138 ibi_s3_bs1 = []
139 ibi_s3_st1 = []
140 ibi_s3_rc1 = []
141 ibi_s3_bs2 = []
142 ibi_s3_st2 = []
143 ibi_s3_rc2 = []
144
145 hr_s3_bs1 = []
146 hr_s3_st1 = []
147 hr_s3_rc1 = []
148 hr_s3_bs2 = []
149 hr_s3_st2 = []
150 hr_s3_rc2 = []
151
152 bvp_e4_bs1 = []
153 bvp_e4_st1 = []
154 bvp_e4_rc1 = []
155 bvp_e4_bs2 = []
156 bvp_e4_st2 = []
157 bvp_e4_rc2 = []
158
159 ibi_e4_bs1 = []
160 ibi_e4_st1 = []
161 ibi_e4_rc1 = []
162 ibi_e4_bs2 = []
163 ibi_e4_st2 = []
164 ibi_e4_rc2 = []
165
166 hr_e4_bs1 = []
167 hr_e4_st1 = []
168 hr_e4_rc1 = []
169 hr_e4_bs2 = []
170 hr_e4_st2 = []
171 hr_e4_rc2 = []
172
173 tppg_s3_bs1 = []
174 tppg_s3_st1 = []
175 tppg_s3_rc1 = []
176 tppg_s3_bs2 = []
177 tppg_s3_st2 = []
178 tppg_s3_rc2 = []
179
180 tibi_s3_bs1 = []
181 tibi_s3_st1 = []
182 tibi_s3_rc1 = []
183 tibi_s3_bs2 = []
184 tibi_s3_st2 = []
185 tibi_s3_rc2 = []
186
187 thr_s3_bs1 = []
188 thr_s3_st1 = []
189 thr_s3_rc1 = []

```


VALIDATION OF A WRIST-WORN PPG SENSOR

```

190 thr_s3_bs2 = []
191 thr_s3_st2 = []
192 thr_s3_rc2 = []
193
194 tbvp_e4_bs1 = []
195 tbvp_e4_st1 = []
196 tbvp_e4_rc1 = []
197 tbvp_e4_bs2 = []
198 tbvp_e4_st2 = []
199 tbvp_e4_rc2 = []
200
201 tibi_e4_bs1 = []
202 tibi_e4_st1 = []
203 tibi_e4_rc1 = []
204 tibi_e4_bs2 = []
205 tibi_e4_st2 = []
206 tibi_e4_rc2 = []
207
208 thr_e4_bs1 = []
209 thr_e4_st1 = []
210 thr_e4_rc1 = []
211 thr_e4_bs2 = []
212 thr_e4_st2 = []
213 thr_e4_rc2 = []
214
215 subjects = []
216 bpm_e4 = []
217 bpm_s3 = []
218 ibi_e4 = []
219 ibi_s3 = []
220 rmssd_e4 = []
221 rmssd_s3 = []
222 sdnns_e4 = []
223 sdnns_s3 = []
224
225 zero_ibi_bs1 = False
226 zero_ibi_st1 = False
227 zero_ibi_rc1 = False
228 zero_ibi_bs2 = False
229 zero_ibi_st2 = False
230 zero_ibi_rc2 = False
231
232 zero_hr_bs1 = False
233 zero_hr_st1 = False
234 zero_hr_rc1 = False
235 zero_hr_bs2 = False
236 zero_hr_st2 = False
237 zero_hr_rc2 = False
238
239 corr_coef = []
240 corr_mode = []
241
242 modes = [ 'Modes' ]
243 bpm = [ 'Bpm' ]
244 breathingrate = [ 'Breathingrate' ]
245 hf = [ 'Hf' ]
246 hr_mad = [ 'Hr_mad' ]
247 ibi = [ 'Ibi' ]
248 lf = [ 'Lf' ]
249 lf_hf = [ 'Lf/Hf' ]
250 pnn20 = [ 'Pnn20' ]
251 pnn50 = [ 'Pnn50' ]
252 rmssd = [ 'Rmssd' ]

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

253 s          = ['S']
254 sd1         = ['Sd1']
255 sd1_sd2     = ['Sd1/Sd2']
256 sd2         = ['Sd2']
257 sdn         = ['Sdnn']
258 sds         = ['Sdsd']
259
260 subjects = 19
261
262 #FUNCTIONS
263
264 def get_s3_units(list):
265     unit = list[0]
266     #print('Unit: ' + unit)
267
268     time_units.append(unit)
269
270     adj_list = np.delete(list, 0)
271
272     return adj_list
273
274 def get_bvp_timestamp(list):
275     timestamp = list[0]
276     print('Timestamp: ' + str(timestamp))
277
278     #timestamps_bvp.append(timestamp)
279
280     adj_list = np.delete(list, 0)
281     adj_list = np.delete(list, 0)
282
283     return adj_list
284
285 def get_bvp_length(subj):
286     data_path = 'p_' + str(subj) + '_BVP.csv'
287     print('CurrentFile: ' + data_path)
288
289     bvp_e4_raw = hp.get_data(data_path, column_name = 'BVP')
290     bvp_e4_adj = get_bvp_timestamp(bvp_e4_raw)
291
292     y = len(bvp_e4_adj)
293
294     print(str(data_path) + ' Length: ' + str(y))
295
296 def downsampling_check(data_list, time_list):
297     a = len(data_list)
298     b = len(time_list)
299
300     if a == b:
301         print('Successful DownSampling: ' + str(a))
302     else:
303         print('Something went Wrong: ' + str(a) + ' ' + str(b))
304
305 def shorten_timestamps(old_list, new_list):
306     for x in range(0, len(old_list)):
307         y = old_list[x]
308         z = int(str(y)[:10])
309         #print(z)
310
311         new_list.append(z)
312
313     return new_list
314
315 def read_bvp_data(subj):

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

316     data_path = 'p_' + str(subj) + '_BVP.csv'
317     print('CurrentFile: ' + data_path)
318
319     bvp_e4_raw = hp.get_data(data_path, column_name = 'BVP')
320     bvp_e4_adj = get_bvp_timestamp(bvp_e4_raw)
321
322     return bvp_e4_adj
323
324 def read_ibi_data(subj):
325     data_path = 'p_' + str(subj) + '_IBI.csv'
326     print('CurrentFile: ' + data_path)
327
328     ibi_e4_adj = hp.get_data(data_path, column_name = "IBI")
329
330     ibt_e4_adj = hp.get_data(data_path, column_name = "Time")
331
332     return ibi_e4_adj, ibt_e4_adj
333
334 def read_hr_data(subj):
335     data_path = 'p_' + str(subj) + '_HR.csv'
336     print('CurrentFile: ' + data_path)
337
338     hr_e4_raw = hp.get_data(data_path, column_name = "HR")
339     hr_e4_adj = hr_e4_raw
340
341     return hr_e4_adj
342
343 def create_bvp_time(subj, bvp_e4_adj, timestamps_bvp):
344     length = len(bvp_e4_adj)
345     act_length = length - 1
346
347     bvp_time = []
348     bvp_time.append(timestamps_bvp[subj])
349
350     counter = timestamps_bvp[subj]
351
352     for x in range(1, (act_length + 1)):
353         counter += 16
354         bvp_time.append(counter)
355
356     if length == len(bvp_time):
357         print('BVP_TIME CREATION SUCCESSFUL!')
358         print(length)
359         print(len(bvp_time))
360
361     return bvp_time
362
363 def create_ibi_time(subj, ibi_e4_adj, timestamps_ibi):
364     length = len(ibi_e4_adj)
365
366     ibi_time = []
367     counter = timestamps_ibi[subj]
368
369     for x in range(0, length):
370         y = counter + (ibi_e4_adj[x] * 1000)
371         ibi_time.append(y)
372
373     if length == len(ibi_time):
374         print('IBI_TIME CREATION SUCCESSFUL!')
375         print(length)
376         print(len(ibi_time))
377
378     return ibi_time

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

379
380 def create_hr_time(subj, hr_e4_adj, timestamps_hr):
381     length = len(hr_e4_adj)
382     act_length = length - 1
383
384     hr_time = []
385     hr_time.append(timestamps_hr[subj])
386
387     counter = timestamps_hr[subj]
388
389     for x in range(1, (act_length + 1)):
390         counter += 1000
391         hr_time.append(counter)
392
393     if length == len(hr_time):
394         print('HR.TIME CREATION SUCCESSFUL!')
395         print(length)
396         print(len(hr_time))
397
398     return hr_time
399
400 def read_s3_data(subj, time_s3_adj, ppg_s3_adj, ibi_s3_adj, hr_s3_adj):
401     data_path = 'p-' + str(subj) + '_Shimmer.csv'
402     print('CurrentFile: ' + data_path)
403
404     time_s3_raw = hp.get_data(data_path, column_name = 'Shimmer_CD26_Timestamp_Unix_CAL')
405     time_s3_adj = get_s3_units(time_s3_raw)
406     #print(time_s3_adj)
407
408     ppg_s3_raw = hp.get_data(data_path, column_name = 'Shimmer_CD26_PPG_A13_CAL')
409     ppg_s3_adj = get_s3_units(ppg_s3_raw)
410     #print(ppg_s3_adj)
411
412     ibi_s3_raw = hp.get_data(data_path, column_name = 'Shimmer_CD26_PPG_IBI_CAL')
413     ibi_s3_adj = get_s3_units(ibi_s3_raw)
414     #print(ibi_s3_adj)
415
416     hr_s3_raw = hp.get_data(data_path, column_name = 'Shimmer_CD26_PPGtoHR_CAL')
417     hr_s3_adj = get_s3_units(hr_s3_raw)
418     #print(hr_s3_adj)
419
420     return time_s3_adj, ppg_s3_adj, ibi_s3_adj, hr_s3_adj
421
422 def split_up_s3_ppg(subj, tppg_s3_adj, ppg_s3_adj, ppg_s3_bs1, ppg_s3_st1, \
423                     ppg_s3_rc1, ppg_s3_bs2, ppg_s3_st2, ppg_s3_rc2, tppg_s3_bs1, \
424                     tppg_s3_st1, tppg_s3_rc1, tppg_s3_bs2, tppg_s3_st2, tppg_s3_rc2):
425     ppg_s3_length = len(ppg_s3_adj)
426     print(ppg_s3_length)
427
428     for y in range(0, ppg_s3_length):
429         i = float(tppg_s3_adj[y])
430
431         #BS1
432         if i >= float(start_time_bs1[subj]) and i <= float(end_time_bs1[subj]):
433             ppg_s3_bs1.append(ppg_s3_adj[y])
434             tppg_s3_bs1.append(tppg_s3_adj[y])
435
436         #ST1
437         if i >= float(start_time_st1[subj]) and i <= float(end_time_st1[subj]):
438             ppg_s3_st1.append(ppg_s3_adj[y])
439             tppg_s3_st1.append(tppg_s3_adj[y])
440
441         #RC1

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

442         if i >= float(start_time_rc1[subj]) and i <= float(end_time_rc1[subj]):
443             ppg_s3_rc1.append(ppg_s3_adj[y])
444             tppg_s3_rc1.append(tppg_s3_adj[y])
445
446         #BS2
447         if i >= float(start_time_bs2[subj]) and i <= float(end_time_bs2[subj]):
448             ppg_s3_bs2.append(ppg_s3_adj[y])
449             tppg_s3_bs2.append(tppg_s3_adj[y])
450
451         #ST2
452         if i >= float(start_time_st2[subj]) and i <= float(end_time_st2[subj]):
453             ppg_s3_st2.append(ppg_s3_adj[y])
454             tppg_s3_st2.append(tppg_s3_adj[y])
455
456         #RC2
457         if i >= float(start_time_rc2[subj]) and i <= float(end_time_rc2[subj]):
458             ppg_s3_rc2.append(ppg_s3_adj[y])
459             tppg_s3_rc2.append(tppg_s3_adj[y])
460
461     print('SUBJECT: ' + str(subj))
462     print('S3_PPG')
463     print('TPPG_BS1.Length: ' + str(len(tppg_s3_bs1)))
464     print('PPG_BS1.Length: ' + str(len(ppg_s3_bs1)))
465
466     print('TPPG_ST1.Length: ' + str(len(tppg_s3_st1)))
467     print('PPG_ST1.Length: ' + str(len(ppg_s3_st1)))
468
469     print('TPPG_RC1.Length: ' + str(len(tppg_s3_rc1)))
470     print('PPG_RC1.Length: ' + str(len(ppg_s3_rc1)))
471
472     print('TPPG_BS2.Length: ' + str(len(tppg_s3_bs2)))
473     print('PPG_BS2.Length: ' + str(len(ppg_s3_bs2)))
474
475     print('TPPG_ST2.Length: ' + str(len(tppg_s3_st2)))
476     print('PPG_ST2.Length: ' + str(len(ppg_s3_st2)))
477
478     print('TPPG_RC2.Length: ' + str(len(tppg_s3_rc2)))
479     print('PPG_RC2.Length: ' + str(len(ppg_s3_rc2)))
480
481     return ppg_s3_bs1, ppg_s3_st1, ppg_s3_rc1, ppg_s3_bs2, ppg_s3_st2, ppg_s3_rc2,
482           tppg_s3_bs1, tppg_s3_st1, tppg_s3_rc1, tppg_s3_bs2, tppg_s3_st2, tppg_s3_rc2
483
484 def split_up_s3_ibi(subj, tibi_s3_adj, ibi_s3_adj, ibi_s3_bs1, ibi_s3_st1, \
485                  ibi_s3_rc1, ibi_s3_bs2, ibi_s3_st2, ibi_s3_rc2, tibi_s3_bs1, \
486                  tibi_s3_st1, tibi_s3_rc1, tibi_s3_bs2, tibi_s3_st2, tibi_s3_rc2):
487     ibi_s3_length = len(ibi_s3_adj)
488     print(ibi_s3_length)
489
490     for y in range(0, ibi_s3_length):
491         i = float(tibi_s3_adj[y])
492
493         #BS1
494         if i >= float(start_time_bs1[subj]) and i <= float(end_time_bs1[subj]):
495             ibi_s3_bs1.append(ibi_s3_adj[y])
496             tibi_s3_bs1.append(tibi_s3_adj[y])
497
498         #ST1
499         if i >= float(start_time_st1[subj]) and i <= float(end_time_st1[subj]):
500             ibi_s3_st1.append(ibi_s3_adj[y])
501             tibi_s3_st1.append(tibi_s3_adj[y])
502
503         #RC1
504         if i >= float(start_time_rc1[subj]) and i <= float(end_time_rc1[subj]):

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

504         ibi_s3_rc1.append(ibi_s3_adj[y])
505         tibi_s3_rc1.append(tibi_s3_adj[y])
506
507     #BS2
508     if i >= float(start_time_bs2[subj]) and i <= float(end_time_bs2[subj]):
509         ibi_s3_bs2.append(ibi_s3_adj[y])
510         tibi_s3_bs2.append(tibi_s3_adj[y])
511
512     #ST2
513     if i >= float(start_time_st2[subj]) and i <= float(end_time_st2[subj]):
514         ibi_s3_st2.append(ibi_s3_adj[y])
515         tibi_s3_st2.append(tibi_s3_adj[y])
516
517     #RC2
518     if i >= float(start_time_rc2[subj]) and i <= float(end_time_rc2[subj]):
519         ibi_s3_rc2.append(ibi_s3_adj[y])
520         tibi_s3_rc2.append(tibi_s3_adj[y])
521
522     print('SUBJECT: ' + str(subj))
523     print('S3_IBI')
524     print('TIBI_BS1_Length: ' + str(len(tibi_s3_bs1)))
525     print('IBI_BS1_Length: ' + str(len(ibi_s3_bs1)))
526
527     print('TIBI_ST1_Length: ' + str(len(tibi_s3_st1)))
528     print('IBI_ST1_Length: ' + str(len(ibi_s3_st1)))
529
530     print('TIBI_RC1_Length: ' + str(len(tibi_s3_rc1)))
531     print('IBI_RC1_Length: ' + str(len(ibi_s3_rc1)))
532
533     print('TIBI_BS2_Length: ' + str(len(tibi_s3_bs2)))
534     print('IBI_BS2_Length: ' + str(len(ibi_s3_bs2)))
535
536     print('TIBI_ST2_Length: ' + str(len(tibi_s3_st2)))
537     print('IBI_ST2_Length: ' + str(len(ibi_s3_st2)))
538
539     print('TIBI_RC2_Length: ' + str(len(tibi_s3_rc2)))
540     print('IBI_RC2_Length: ' + str(len(ibi_s3_rc2)))
541
542     return ibi_s3_bs1, ibi_s3_st1, ibi_s3_rc1, ibi_s3_bs2, ibi_s3_st2, ibi_s3_rc2,
543            tibi_s3_bs1, tibi_s3_st1, tibi_s3_rc1, tibi_s3_bs2, tibi_s3_st2, tibi_s3_rc2
544
545 def split_up_s3_hr(subj, thr_s3_adj, hr_s3_adj, hr_s3_bs1, hr_s3_st1, hr_s3_rc1, \
546                  hr_s3_bs2, hr_s3_st2, hr_s3_rc2, thr_s3_bs1, thr_s3_st1, \
547                  thr_s3_rc1, thr_s3_bs2, thr_s3_st2, thr_s3_rc2):
548     hr_s3_length = len(hr_s3_adj)
549     print(hr_s3_length)
550
551     for y in range(0, hr_s3_length):
552         i = float(thr_s3_adj[y])
553
554         #BS1
555         if i >= float(start_time_bs1[subj]) and i <= float(end_time_bs1[subj]):
556             hr_s3_bs1.append(hr_s3_adj[y])
557             thr_s3_bs1.append(thr_s3_adj[y])
558
559         #ST1
560         if i >= float(start_time_st1[subj]) and i <= float(end_time_st1[subj]):
561             hr_s3_st1.append(hr_s3_adj[y])
562             thr_s3_st1.append(thr_s3_adj[y])
563
564         #RC1
565         if i >= float(start_time_rc1[subj]) and i <= float(end_time_rc1[subj]):
566             hr_s3_rc1.append(hr_s3_adj[y])

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

566         thr_s3_rc1.append(thr_s3_adj[y])
567
568     #BS2
569     if i >= float(start_time_bs2[subj]) and i <= float(end_time_bs2[subj]):
570         hr_s3_bs2.append(hr_s3_adj[y])
571         thr_s3_bs2.append(thr_s3_adj[y])
572
573     #ST2
574     if i >= float(start_time_st2[subj]) and i <= float(end_time_st2[subj]):
575         hr_s3_st2.append(hr_s3_adj[y])
576         thr_s3_st2.append(thr_s3_adj[y])
577
578     #RC2
579     if i >= float(start_time_rc2[subj]) and i <= float(end_time_rc2[subj]):
580         hr_s3_rc2.append(hr_s3_adj[y])
581         thr_s3_rc2.append(thr_s3_adj[y])
582
583     print('SUBJECT: ' + str(subj))
584     print('S3_HR')
585     print('THR_BS1.Length: ' + str(len(thr_s3_bs1)))
586     print('HR_BS1.Length: ' + str(len(hr_s3_bs1)))
587
588     print('THR_ST1.Length: ' + str(len(thr_s3_st1)))
589     print('HR_ST1.Length: ' + str(len(hr_s3_st1)))
590
591     print('THR_RC1.Length: ' + str(len(thr_s3_rc1)))
592     print('HR_RC1.Length: ' + str(len(hr_s3_rc1)))
593
594     print('THR_BS2.Length: ' + str(len(thr_s3_bs2)))
595     print('HR_BS2.Length: ' + str(len(hr_s3_bs2)))
596
597     print('THR_ST2.Length: ' + str(len(thr_s3_st2)))
598     print('HR_ST2.Length: ' + str(len(hr_s3_st2)))
599
600     print('THR_RC2.Length: ' + str(len(thr_s3_rc2)))
601     print('HR_RC2.Length: ' + str(len(hr_s3_rc2)))
602
603     return hr_s3_bs1, hr_s3_st1, hr_s3_rc1, hr_s3_bs2, hr_s3_st2, hr_s3_rc2, thr_s3_bs1,
        thr_s3_st1, thr_s3_rc1, thr_s3_bs2, thr_s3_st2, thr_s3_rc2
604
605 def split_up_e4_bvp(subj, bvp_e4_adj, time_e4_bvp, bvp_e4_bs1, bvp_e4_st1, \
606                     bvp_e4_rc1, bvp_e4_bs2, bvp_e4_st2, bvp_e4_rc2, tbvp_e4_bs1, \
607                     tbvp_e4_st1, tbvp_e4_rc1, tbvp_e4_bs2, tbvp_e4_st2, tbvp_e4_rc2):
608     print('SPLITTING OF E4 BVP')
609
610     bvp_e4_length = len(bvp_e4_adj)
611     tbvp_e4_length = len(time_e4_bvp)
612     print(bvp_e4_length)
613     print(tbvp_e4_length)
614
615     for y in range(0, bvp_e4_length):
616         i = float(time_e4_bvp[y])
617
618     #BS1
619     if i >= float(start_time_bs1[subj]) and i <= float(end_time_bs1[subj]):
620         bvp_e4_bs1.append(bvp_e4_adj[y])
621         tbvp_e4_bs1.append(time_e4_bvp[y])
622
623     #ST1
624     if i >= float(start_time_st1[subj]) and i <= float(end_time_st1[subj]):
625         bvp_e4_st1.append(bvp_e4_adj[y])
626         tbvp_e4_st1.append(time_e4_bvp[y])
627

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

628     #RC1
629     if i >= float(start_time_rc1[subj]) and i <= float(end_time_rc1[subj]):
630         bvp_e4_rc1.append(bvp_e4_adj[y])
631         tbvp_e4_rc1.append(time_e4_bvp[y])
632
633     #BS2
634     if i >= float(start_time_bs2[subj]) and i <= float(end_time_bs2[subj]):
635         bvp_e4_bs2.append(bvp_e4_adj[y])
636         tbvp_e4_bs2.append(time_e4_bvp[y])
637
638     #ST2
639     if i >= float(start_time_st2[subj]) and i <= float(end_time_st2[subj]):
640         bvp_e4_st2.append(bvp_e4_adj[y])
641         tbvp_e4_st2.append(time_e4_bvp[y])
642
643     #RC2
644     if i >= float(start_time_rc2[subj]) and i <= float(end_time_rc2[subj]):
645         bvp_e4_rc2.append(bvp_e4_adj[y])
646         tbvp_e4_rc2.append(time_e4_bvp[y])
647
648     print('SUBJECT: ' + str(subj))
649     print('E4_BVP')
650     print('TIME_BS1.Length: ' + str(len(tbvp_e4_bs1)))
651     print('BVP_BS1.Length: ' + str(len(bvp_e4_bs1)))
652
653     print('TIME_ST1.Length: ' + str(len(tbvp_e4_st1)))
654     print('BVP_ST1.Length: ' + str(len(bvp_e4_st1)))
655
656     print('TIME_RC1.Length: ' + str(len(tbvp_e4_rc1)))
657     print('BVP_RC1.Length: ' + str(len(bvp_e4_rc1)))
658
659     print('TIME_BS2.Length: ' + str(len(tbvp_e4_bs2)))
660     print('BVP_BS2.Length: ' + str(len(bvp_e4_bs2)))
661
662     print('TIME_ST2.Length: ' + str(len(tbvp_e4_st2)))
663     print('BVP_ST2.Length: ' + str(len(bvp_e4_st2)))
664
665     print('TIME_RC2.Length: ' + str(len(tbvp_e4_rc2)))
666     print('BVP_RC2.Length: ' + str(len(bvp_e4_rc2)))
667
668     return bvp_e4_bs1, bvp_e4_st1, bvp_e4_rc1, bvp_e4_bs2, bvp_e4_st2, bvp_e4_rc2,
669     tbvp_e4_bs1, tbvp_e4_st1, tbvp_e4_rc1, tbvp_e4_bs2, tbvp_e4_st2, tbvp_e4_rc2
670
671 def split_up_e4_ibi(subj, ibi_e4_adj, time_e4_ibi, ibi_e4_bs1, ibi_e4_st1, \
672                    ibi_e4_rc1, ibi_e4_bs2, ibi_e4_st2, ibi_e4_rc2, tibi_e4_bs1, \
673                    tibi_e4_st1, tibi_e4_rc1, tibi_e4_bs2, tibi_e4_st2, tibi_e4_rc2):
674     ibi_e4_length = len(ibi_e4_adj)
675     print(ibi_e4_length)
676
677     for y in range(0, ibi_e4_length):
678         i = float(time_e4_ibi[y])
679
680         #BS1
681         if i >= float(start_time_bs1[subj]) and i <= float(end_time_bs1[subj]):
682             ibi_e4_bs1.append(ibi_e4_adj[y])
683             tibi_e4_bs1.append(time_e4_ibi[y])
684
685         #ST1
686         if i >= float(start_time_st1[subj]) and i <= float(end_time_st1[subj]):
687             ibi_e4_st1.append(ibi_e4_adj[y])
688             tibi_e4_st1.append(time_e4_ibi[y])
689
690         #RC1

```


VALIDATION OF A WRIST-WORN PPG SENSOR

```

690         if i >= float(start_time_rc1[subj]) and i <= float(end_time_rc1[subj]):
691             ibi_e4_rc1.append(ibi_e4_adj[y])
692             tibi_e4_rc1.append(time_e4_ibi[y])
693
694         #BS2
695         if i >= float(start_time_bs2[subj]) and i <= float(end_time_bs2[subj]):
696             ibi_e4_bs2.append(ibi_e4_adj[y])
697             tibi_e4_bs2.append(time_e4_ibi[y])
698
699         #ST2
700         if i >= float(start_time_st2[subj]) and i <= float(end_time_st2[subj]):
701             ibi_e4_st2.append(ibi_e4_adj[y])
702             tibi_e4_st2.append(time_e4_ibi[y])
703
704         #RC2
705         if i >= float(start_time_rc2[subj]) and i <= float(end_time_rc2[subj]):
706             ibi_e4_rc2.append(ibi_e4_adj[y])
707             tibi_e4_rc2.append(time_e4_ibi[y])
708
709     print('SUBJECT: ' + str(subj))
710     print('E4_IBI')
711     print('TIME_BS1.Length: ' + str(len(tibi_e4_bs1)))
712     print('IBI_BS1.Length: ' + str(len(ibi_e4_bs1)))
713
714     print('TIME_ST1.Length: ' + str(len(tibi_e4_st1)))
715     print('IBI_ST1.Length: ' + str(len(ibi_e4_st1)))
716
717     print('TIME_RC1.Length: ' + str(len(tibi_e4_rc1)))
718     print('IBI_RC1.Length: ' + str(len(ibi_e4_rc1)))
719
720     print('TIME_BS2.Length: ' + str(len(tibi_e4_bs2)))
721     print('IBI_BS2.Length: ' + str(len(ibi_e4_bs2)))
722
723     print('TIME_ST2.Length: ' + str(len(tibi_e4_st2)))
724     print('IBI_ST2.Length: ' + str(len(ibi_e4_st2)))
725
726     print('TIME_RC2.Length: ' + str(len(tibi_e4_rc2)))
727     print('IBI_RC2.Length: ' + str(len(ibi_e4_rc2)))
728
729     return ibi_e4_bs1, ibi_e4_st1, ibi_e4_rc1, ibi_e4_bs2, ibi_e4_st2, ibi_e4_rc2,
730            tibi_e4_bs1, tibi_e4_st1, tibi_e4_rc1, tibi_e4_bs2, tibi_e4_st2, tibi_e4_rc2
731
732 def split_up_e4_hr(subj, hr_e4_adj, time_e4_hr, hr_e4_bs1, hr_e4_st1, hr_e4_rc1, \
733                  hr_e4_bs2, hr_e4_st2, hr_e4_rc2, thr_e4_bs1, thr_e4_st1, \
734                  thr_e4_rc1, thr_e4_bs2, thr_e4_st2, thr_e4_rc2):
735     hr_e4_length = len(hr_e4_adj)
736     print(hr_e4_length)
737
738     for y in range(0, hr_e4_length):
739         i = float(time_e4_hr[y])
740
741         #BS1
742         if i >= float(start_time_bs1[subj]) and i <= float(end_time_bs1[subj]):
743             hr_e4_bs1.append(hr_e4_adj[y])
744             thr_e4_bs1.append(time_e4_hr[y])
745
746         #ST1
747         if i >= float(start_time_st1[subj]) and i <= float(end_time_st1[subj]):
748             hr_e4_st1.append(hr_e4_adj[y])
749             thr_e4_st1.append(time_e4_hr[y])
750
751         #RC1
752         if i >= float(start_time_rc1[subj]) and i <= float(end_time_rc1[subj]):

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

752         hr_e4_rc1.append(hr_e4_adj[y])
753         thr_e4_rc1.append(time_e4_hr[y])
754
755     #BS2
756     if i >= float(start_time_bs2[subj]) and i <= float(end_time_bs2[subj]):
757         hr_e4_bs2.append(hr_e4_adj[y])
758         thr_e4_bs2.append(time_e4_hr[y])
759
760     #ST2
761     if i >= float(start_time_st2[subj]) and i <= float(end_time_st2[subj]):
762         hr_e4_st2.append(hr_e4_adj[y])
763         thr_e4_st2.append(time_e4_hr[y])
764
765     #RC2
766     if i >= float(start_time_rc2[subj]) and i <= float(end_time_rc2[subj]):
767         hr_e4_rc2.append(hr_e4_adj[y])
768         thr_e4_rc2.append(time_e4_hr[y])
769
770     print('SUBJECT: ' + str(subj))
771     print('E4.HR')
772     print('TIME_BS1.Length: ' + str(len(thr_e4_bs1)))
773     print('HR_BS1.Length: ' + str(len(hr_e4_bs1)))
774
775     print('TIME_ST1.Length: ' + str(len(thr_e4_st1)))
776     print('HR_ST1.Length: ' + str(len(hr_e4_st1)))
777
778     print('TIME_RC1.Length: ' + str(len(thr_e4_rc1)))
779     print('HR_RC1.Length: ' + str(len(hr_e4_rc1)))
780
781     print('TIME_BS2.Length: ' + str(len(thr_e4_bs2)))
782     print('HR_BS2.Length: ' + str(len(hr_e4_bs2)))
783
784     print('TIME_ST2.Length: ' + str(len(thr_e4_st2)))
785     print('HR_ST2.Length: ' + str(len(hr_e4_st2)))
786
787     print('TIME_RC2.Length: ' + str(len(thr_e4_rc2)))
788     print('HR_RC2.Length: ' + str(len(hr_e4_rc2)))
789
790     return hr_e4_bs1, hr_e4_st1, hr_e4_rc1, hr_e4_bs2, hr_e4_st2, hr_e4_rc2, thr_e4_bs1,
thr_e4_st1, thr_e4_rc1, thr_e4_bs2, thr_e4_st2, thr_e4_rc2
791
792 def cut_in_shape(subj, time_s3_adj, ppg_s3_adj, ibi_s3_adj, hr_s3_adj, bvp_e4_adj, \
793                 ibi_e4_adj, hr_e4_adj, time_e4_bvp, time_e4_ibi, time_e4_hr):
794     tppg_s3 = []
795     tibi_s3 = []
796     thr_s3 = []
797     ppg_s3 = []
798     ibi_s3 = []
799     hr_s3 = []
800
801     s3_length = len(time_s3_adj)
802     print(s3_length)
803
804     for y in range(0, s3_length):
805         i = float(time_s3_adj[y])
806
807         if i >= float(start_time_bs1[subj]) and i <= float(end_time_rc2[subj]):
808             tppg_s3.append(time_s3_adj[y])
809             tibi_s3.append(time_s3_adj[y])
810             thr_s3.append(time_s3_adj[y])
811
812             ppg_s3.append(ppg_s3_adj[y])
813             ibi_s3.append(ibi_s3_adj[y])

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

814         hr_s3.append(hr_s3_adj[y])
815
816     bvp_e4 = []
817     ibi_e4 = []
818     hr_e4 = []
819     tbvp_e4 = []
820     tibi_e4 = []
821     thr_e4 = []
822
823     e4_bvp_length = len(bvp_e4_adj)
824     print('BVP LENGTH: ' + str(e4_bvp_length))
825
826     g_rest = []
827     g_trest = []
828     g_pos = []
829     k_rest = []
830     k_trest = []
831     k_pos = []
832
833     for y in range(0, e4_bvp_length):
834         i = float(time_e4_bvp[y])
835
836         if i >= float(start_time_bs1[subj]) and i <= float(end_time_rc2[subj]):
837             tbvp_e4.append(time_e4_bvp[y])
838             bvp_e4.append(bvp_e4_adj[y])
839         if i < float(start_time_bs1[subj]):
840             k_rest.append(bvp_e4_adj[y])
841             k_trest.append(time_e4_bvp[y])
842             k_pos.append(y)
843         if i > float(end_time_rc2[subj]):
844             g_rest.append(bvp_e4_adj[y])
845             g_trest.append(time_e4_bvp[y])
846             g_pos.append(y)
847
848     print('G-REST: ' + str(len(g_rest)))
849     print('G-TREST: ' + str(len(g_trest)))
850     #print(g_pos)
851
852     print('K-REST: ' + str(len(k_rest)))
853     print('K-TREST: ' + str(len(k_trest)))
854     #print(k_pos)
855
856
857     e4_ibi_length = len(ibi_e4_adj)
858     print(e4_ibi_length)
859
860     for y in range(0, e4_ibi_length):
861         i = float(time_e4_ibi[y])
862
863         if i >= float(start_time_bs1[subj]) and i <= float(end_time_rc2[subj]):
864             tibi_e4.append(time_e4_ibi[y])
865             ibi_e4.append(ibi_e4_adj[y])
866
867     e4_hr_length = len(hr_e4_adj)
868     print(e4_hr_length)
869
870     for y in range(0, e4_hr_length):
871         i = float(time_e4_hr[y])
872
873         if i >= float(start_time_bs1[subj]) and i <= float(end_time_rc2[subj]):
874             thr_e4.append(time_e4_hr[y])
875             hr_e4.append(hr_e4_adj[y])
876

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

877     return tppg_s3, tibi_s3, thr_s3, ppg_s3, ibi_s3, hr_s3, bvp_e4, ibi_e4, hr_e4, tbvp_e4,
      tibi_e4, thr_e4
878
879 def downsampling_ppg(subj, tppg_s3_bs1, tppg_s3_st1, tppg_s3_rc1, tppg_s3_bs2, \
880                       tppg_s3_st2, tppg_s3_rc2, bvp_e4_bs1, bvp_e4_st1, bvp_e4_rc1, \
881                       bvp_e4_bs2, bvp_e4_st2, bvp_e4_rc2, ppg_s3_bs1, ppg_s3_st1, \
882                       ppg_s3_rc1, ppg_s3_bs2, ppg_s3_st2, ppg_s3_rc2):
883     bs1_length = len(bvp_e4_bs1)
884     st1_length = len(bvp_e4_st1)
885     rc1_length = len(bvp_e4_rc1)
886     bs2_length = len(bvp_e4_bs2)
887     st2_length = len(bvp_e4_st2)
888     rc2_length = len(bvp_e4_rc2)
889
890     bs1 = ppg_s3_bs1
891     st1 = ppg_s3_st1
892     rc1 = ppg_s3_rc1
893     bs2 = ppg_s3_bs2
894     st2 = ppg_s3_st2
895     rc2 = ppg_s3_rc2
896
897     tbs1 = np.asarray(tppg_s3_bs1, dtype = np.float64)
898     tst1 = np.asarray(tppg_s3_st1, dtype = np.float64)
899     trc1 = np.asarray(tppg_s3_rc1, dtype = np.float64)
900     tbs2 = np.asarray(tppg_s3_bs2, dtype = np.float64)
901     tst2 = np.asarray(tppg_s3_st2, dtype = np.float64)
902     trc2 = np.asarray(tppg_s3_rc2, dtype = np.float64)
903
904     ppg_s3_bs1 = sp.signal.resample(bs1, bs1_length, t = tbs1)
905     ppg_s3_st1 = sp.signal.resample(st1, st1_length, t = tst1)
906     ppg_s3_rc1 = sp.signal.resample(rc1, rc1_length, t = trc1)
907     ppg_s3_bs2 = sp.signal.resample(bs2, bs2_length, t = tbs2)
908     ppg_s3_st2 = sp.signal.resample(st2, st2_length, t = tst2)
909     ppg_s3_rc2 = sp.signal.resample(rc2, rc2_length, t = trc2)
910
911     print('SUBJECT: ' + str(subj))
912     print('PPG-S3')
913     downsampling_check(ppg_s3_bs1[0], bvp_e4_bs1)
914     downsampling_check(ppg_s3_st1[0], bvp_e4_st1)
915     downsampling_check(ppg_s3_rc1[0], bvp_e4_rc1)
916     downsampling_check(ppg_s3_bs2[0], bvp_e4_bs2)
917     downsampling_check(ppg_s3_st2[0], bvp_e4_st2)
918     downsampling_check(ppg_s3_rc2[0], bvp_e4_rc2)
919     print('TPPG-S3')
920     downsampling_check(ppg_s3_bs1[1], bvp_e4_bs1)
921     downsampling_check(ppg_s3_st1[1], bvp_e4_st1)
922     downsampling_check(ppg_s3_rc1[1], bvp_e4_rc1)
923     downsampling_check(ppg_s3_bs2[1], bvp_e4_bs2)
924     downsampling_check(ppg_s3_st2[1], bvp_e4_st2)
925     downsampling_check(ppg_s3_rc2[1], bvp_e4_rc2)
926
927     return ppg_s3_bs1, ppg_s3_st1, ppg_s3_rc1, ppg_s3_bs2, ppg_s3_st2, ppg_s3_rc2
928
929 def downsampling_ibi(subj, zero_ibi_bs1, zero_ibi_st1, zero_ibi_rc1, zero_ibi_bs2, \
930                     zero_ibi_st2, zero_ibi_rc2, tibi_s3_bs1, tibi_s3_st1, \
931                     tibi_s3_rc1, tibi_s3_bs2, tibi_s3_st2, tibi_s3_rc2, \
932                     ibi_e4_bs1, ibi_e4_st1, ibi_e4_rc1, ibi_e4_bs2, ibi_e4_st2, \
933                     ibi_e4_rc2, ibi_s3_bs1, ibi_s3_st1, ibi_s3_rc1, ibi_s3_bs2, \
934                     ibi_s3_st2, ibi_s3_rc2):
935     bs1_length = len(ibi_e4_bs1)
936     st1_length = len(ibi_e4_st1)
937     rc1_length = len(ibi_e4_rc1)
938     bs2_length = len(ibi_e4_bs2)

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

939     st2_length = len(ibi_e4_st2)
940     rc2_length = len(ibi_e4_rc2)
941
942     tbs1 = np.asarray(tibi_s3_bs1, dtype = np.float64)
943     tst1 = np.asarray(tibi_s3_st1, dtype = np.float64)
944     trc1 = np.asarray(tibi_s3_rc1, dtype = np.float64)
945     tbs2 = np.asarray(tibi_s3_bs2, dtype = np.float64)
946     tst2 = np.asarray(tibi_s3_st2, dtype = np.float64)
947     trc2 = np.asarray(tibi_s3_rc2, dtype = np.float64)
948
949     print('SUBJECT: ' + str(subj))
950     print('IBI-S3')
951
952     if bs1_length <= 1:
953         zero_ibi_bs1 = True
954     else:
955         bs1 = ibi_s3_bs1
956         ibi_s3_bs1 = sp.signal.resample(bs1, bs1_length, t = tbs1)
957         downsampling_check(ibi_s3_bs1[0], ibi_e4_bs1)
958         downsampling_check(ibi_s3_bs1[1], ibi_e4_bs1)
959
960     if st1_length <= 1:
961         zero_ibi_st1 = True
962     else:
963         st1 = ibi_s3_st1
964         ibi_s3_st1 = sp.signal.resample(st1, st1_length, t = tst1)
965         downsampling_check(ibi_s3_st1[0], ibi_e4_st1)
966         downsampling_check(ibi_s3_st1[1], ibi_e4_st1)
967
968     if rc1_length <= 1:
969         zero_ibi_rc1 = True
970     else:
971         rc1 = ibi_s3_rc1
972         ibi_s3_rc1 = sp.signal.resample(rc1, rc1_length, t = trc1)
973         downsampling_check(ibi_s3_rc1[0], ibi_e4_rc1)
974         downsampling_check(ibi_s3_rc1[1], ibi_e4_rc1)
975
976     if bs2_length <= 1:
977         zero_ibi_bs2 = True
978     else:
979         bs2 = ibi_s3_bs2
980         ibi_s3_bs2 = sp.signal.resample(bs2, bs2_length, t = tbs2)
981         downsampling_check(ibi_s3_bs2[0], ibi_e4_bs2)
982         downsampling_check(ibi_s3_bs2[1], ibi_e4_bs2)
983
984     if st2_length <= 1:
985         zero_ibi_st2 = True
986     else:
987         st2 = ibi_s3_st2
988         ibi_s3_st2 = sp.signal.resample(st2, st2_length, t = tst2)
989         downsampling_check(ibi_s3_st2[0], ibi_e4_st2)
990         downsampling_check(ibi_s3_st2[1], ibi_e4_st2)
991
992     if rc2_length <= 1:
993         zero_ibi_rc2 = True
994     else:
995         rc2 = ibi_s3_rc2
996         ibi_s3_rc2 = sp.signal.resample(rc2, rc2_length, t = trc2)
997         downsampling_check(ibi_s3_rc2[0], ibi_e4_rc2)
998         downsampling_check(ibi_s3_rc2[1], ibi_e4_rc2)
999
1000     return ibi_s3_bs1, ibi_s3_st1, ibi_s3_rc1, ibi_s3_bs2, ibi_s3_st2, ibi_s3_rc2,
        zero_ibi_bs1, zero_ibi_st1, zero_ibi_rc1, zero_ibi_bs2, zero_ibi_st2, zero_ibi_rc2

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1001
1002 def downsampling_hr(subj, zero_hr_bs1, zero_hr_st1, zero_hr_rc1, zero_hr_bs2, \
1003                     zero_hr_st2, zero_hr_rc2, thr_s3_bs1, thr_s3_st1, thr_s3_rc1, \
1004                     thr_s3_bs2, thr_s3_st2, thr_s3_rc2, hr_e4_bs1, hr_e4_st1, \
1005                     hr_e4_rc1, hr_e4_bs2, hr_e4_st2, hr_e4_rc2, hr_s3_bs1, \
1006                     hr_s3_st1, hr_s3_rc1, hr_s3_bs2, hr_s3_st2, hr_s3_rc2):
1007     bs1_length = len(hr_e4_bs1)
1008     st1_length = len(hr_e4_st1)
1009     rc1_length = len(hr_e4_rc1)
1010     bs2_length = len(hr_e4_bs2)
1011     st2_length = len(hr_e4_st2)
1012     rc2_length = len(hr_e4_rc2)
1013
1014     tbs1 = np.asarray(thr_s3_bs1, dtype = np.float64)
1015     tst1 = np.asarray(thr_s3_st1, dtype = np.float64)
1016     trc1 = np.asarray(thr_s3_rc1, dtype = np.float64)
1017     tbs2 = np.asarray(thr_s3_bs2, dtype = np.float64)
1018     tst2 = np.asarray(thr_s3_st2, dtype = np.float64)
1019     trc2 = np.asarray(thr_s3_rc2, dtype = np.float64)
1020
1021     print('SUBJECT: ' + str(subj))
1022     print('HR_S3')
1023
1024     if bs1_length <= 1:
1025         zero_hr_bs1 = True
1026     else:
1027         bs1 = hr_s3_bs1
1028         hr_s3_bs1 = sp.signal.resample(bs1, bs1_length, t = tbs1)
1029         downsampling_check(hr_s3_bs1[0], hr_e4_bs1)
1030         downsampling_check(hr_s3_bs1[1], hr_e4_bs1)
1031
1032     if st1_length <= 1:
1033         zero_hr_st1 = True
1034     else:
1035         st1 = hr_s3_st1
1036         hr_s3_st1 = sp.signal.resample(st1, st1_length, t = tst1)
1037         downsampling_check(hr_s3_st1[0], hr_e4_st1)
1038         downsampling_check(hr_s3_st1[1], hr_e4_st1)
1039
1040     if rc1_length <= 1:
1041         zero_hr_rc1 = True
1042     else:
1043         rc1 = hr_s3_rc1
1044         hr_s3_rc1 = sp.signal.resample(rc1, rc1_length, t = trc1)
1045         downsampling_check(hr_s3_rc1[0], hr_e4_rc1)
1046         downsampling_check(hr_s3_rc1[1], hr_e4_rc1)
1047
1048     if bs2_length <= 1:
1049         zero_hr_bs2 = True
1050     else:
1051         bs2 = hr_s3_bs2
1052         hr_s3_bs2 = sp.signal.resample(bs2, bs2_length, t = tbs2)
1053         downsampling_check(hr_s3_bs2[0], hr_e4_bs2)
1054         downsampling_check(hr_s3_bs2[1], hr_e4_bs2)
1055
1056     if st2_length <= 1:
1057         zero_hr_st2 = True
1058     else:
1059         st2 = hr_s3_st2
1060         hr_s3_st2 = sp.signal.resample(st2, st2_length, t = tst2)
1061         downsampling_check(hr_s3_st2[0], hr_e4_st2)
1062         downsampling_check(hr_s3_st2[1], hr_e4_st2)
1063

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1064     if rc2_length <= 1:
1065         zero_hr_rc2 = True
1066     else:
1067         rc2 = hr_s3_rc2
1068         hr_s3_rc2 = sp.signal.resample(rc2, rc2_length, t = trc2)
1069         downsampling_check(hr_s3_rc2[0], hr_e4_rc2)
1070         downsampling_check(hr_s3_rc2[1], hr_e4_rc2)
1071
1072     return hr_s3_bs1, hr_s3_st1, hr_s3_rc1, hr_s3_bs2, hr_s3_st2, hr_s3_rc2, zero_hr_bs1,
        zero_hr_st1, zero_hr_rc1, zero_hr_bs2, zero_hr_st2, zero_hr_rc2
1073
1074 def create_plot(x_time, y_data, subj, cond):
1075     plt.plot(x_time, y_data)
1076
1077     title = 'p_' + str(subj) + '_' + cond
1078
1079     plt.title(title)
1080     plt.show()
1081
1082 def create_file(subj, mode, data, tdata):
1083     length = len(data)
1084     subject = []
1085
1086     for x in range(0, length):
1087         subject.append(subj)
1088
1089     rows = zip(subject, tdata, data)
1090
1091     data_path = mode + 'p_' + str(subj) + '.csv'
1092
1093     with open(data_path, "w") as f:
1094         writer = csv.writer(f)
1095         writer.writerow(['SUBJECT', 'TIME', mode])
1096
1097         for row in rows:
1098             writer.writerow(row)
1099
1100     print('Successful FileCreation: ' + mode)
1101
1102 def create_corr_file(mode, data, tdata):
1103     rows = zip(tdata, data)
1104
1105     data_path = mode + '.csv'
1106
1107     with open(data_path, "w") as f:
1108         writer = csv.writer(f)
1109         writer.writerow(['MODE', mode])
1110
1111         for row in rows:
1112             writer.writerow(row)
1113
1114     print('Successful FileCreation: ' + mode)
1115
1116 def np_cross_correlation(subj, mode, xdata, ydata):
1117     title = 'p_' + str(subj) + '_' + mode
1118
1119     corr = np.corrcoef(xdata, ydata)
1120
1121     corr_coef.append(corr[0][1])
1122     corr_mode.append(title)
1123
1124     print(title)
1125     print(corr[0][1])

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1126
1127 def get_hr_s3_parameter(subj, mode, ppg_s3):
1128     tppg = np.asarray(ppg_s3[1], dtype = np.float64)
1129
1130     fs = hp.get_samplerate_mstimer(tppg)
1131
1132     working_data, measures = hp.process(ppg_s3[0], fs, calc_freq = False)
1133
1134     modes.append(mode)
1135     bpm.append(measures['bpm'])
1136     breathingrate.append(measures['breathingrate'])
1137     #hf.append(measures['hf'])
1138     hr_mad.append(measures['hr_mad'])
1139     ibi.append(measures['ibi'])
1140     #lf.append(measures['lf'])
1141     #lf_hf.append(measures['lf/hf'])
1142     pnn20.append(measures['pnn20'])
1143     pnn50.append(measures['pnn50'])
1144     rmssd.append(measures['rmssd'])
1145     s.append(measures['s'])
1146     sd1.append(measures['sd1'])
1147     sd1_sd2.append(measures['sd1/sd2'])
1148     sd2.append(measures['sd2'])
1149     sdnns.append(measures['sdnns'])
1150     sdsd.append(measures['sdsd'])
1151
1152     title = 'Heart Rate Signal Peak Detection ' + mode + '_p' + str(subj)
1153
1154     plt.figure(figsize=(12,4))
1155
1156     plot_object = hp.plotter(working_data, measures, show = False, title = title)
1157
1158     save = mode + '_p' + str(subj) + ' Heart Rate Signal Peak Detection.jpg'
1159
1160     plot_object.savefig(save)
1161     plot_object.show()
1162
1163     print('Successful: p_' + str(subj) + '_' + mode)
1164
1165 def get_hr_e4_parameter(subj, mode, bvp_e4, tbvp_e4):
1166     tbvp = np.asarray(tbvp_e4, dtype = np.float64)
1167     bvp = np.asarray(bvp_e4, dtype = np.float64)
1168
1169     fs = hp.get_samplerate_mstimer(tbvp)
1170
1171     working_data, measures = hp.process(bvp, fs, calc_freq = False)
1172
1173     modes.append(mode)
1174     bpm.append(measures['bpm'])
1175     breathingrate.append(measures['breathingrate'])
1176     #hf.append(measures['hf'])
1177     hr_mad.append(measures['hr_mad'])
1178     ibi.append(measures['ibi'])
1179     #lf.append(measures['lf'])
1180     #lf_hf.append(measures['lf/hf'])
1181     pnn20.append(measures['pnn20'])
1182     pnn50.append(measures['pnn50'])
1183     rmssd.append(measures['rmssd'])
1184     s.append(measures['s'])
1185     sd1.append(measures['sd1'])
1186     sd1_sd2.append(measures['sd1/sd2'])
1187     sd2.append(measures['sd2'])
1188     sdnns.append(measures['sdnns'])

```


VALIDATION OF A WRIST-WORN PPG SENSOR

```

1189     sdsd.append(measures['sdsd'])
1190
1191     title = 'Heart Rate Signal Peak Detection ' + mode + '_p' + str(subj)
1192
1193     plt.figure(figsize=(12,4))
1194
1195     plot_object = hp.plotter(working_data, measures, show = False, title = title)
1196
1197     save = mode + '_p' + str(subj) + ' Heart Rate Signal Peak Detection.jpg'
1198
1199     plot_object.savefig(save)
1200     plot_object.show()
1201
1202     print('Successful: p_' + str(subj) + '_' + mode)
1203
1204 def save_hr_parameters(subj, modes, bpm, breathingrate, hr_mad, ibi, pnn20, \
1205                       pnn50, rmssd, s, sd1, sd1_sd2, sd2, sdnn, sdsd):
1206     rows = zip(modes, bpm, breathingrate, hr_mad, ibi, pnn20, pnn50, rmssd, s, \
1207              sd1, sd1_sd2, sd2, sdnn, sdsd)
1208
1209     print(len(bpm))
1210     print(len(breathingrate))
1211     print(len(hr_mad))
1212     print(len(ibi))
1213     print(len(pnn20))
1214     print(len(pnn50))
1215     print(len(rmssd))
1216
1217     data_path = 'PPG.p_' + str(subj) + '_Hear_Rate_Parameters.csv'
1218
1219     with open(data_path, "w") as f:
1220         writer = csv.writer(f)
1221
1222         for row in rows:
1223             writer.writerow(row)
1224
1225     print('Successful FileCreation: ' + str(data_path))
1226
1227 def read_hr_parameters(subj, bpm_e4, bpm_s3, ibi_e4, ibi_s3, rmssd_e4, rmssd_s3, \
1228                       sdnn_e4, sdnn_s3):
1229     data_path = 'PPG.p_' + str(subj) + '_Hear_Rate_Parameters.csv'
1230     print('CurrentFile: ' + data_path)
1231
1232     subjects.append(subj)
1233
1234     bpm_raw = hp.get_data(data_path, column_name = 'Bpm')
1235     bpm_s3_mean = (bpm_raw[0] + bpm_raw[1] + bpm_raw[2] + \
1236                  bpm_raw[3] + bpm_raw[4] + bpm_raw[5]) / 6
1237     bpm_e4_mean = (bpm_raw[6] + bpm_raw[7] + bpm_raw[8] + \
1238                  bpm_raw[9] + bpm_raw[10] + bpm_raw[11]) / 6
1239
1240     bpm_s3.append(bpm_s3_mean)
1241     bpm_e4.append(bpm_e4_mean)
1242
1243     ibi_raw = hp.get_data(data_path, column_name = 'Ibi')
1244     ibi_s3_mean = (ibi_raw[0] + ibi_raw[1] + ibi_raw[2] + \
1245                  ibi_raw[3] + ibi_raw[4] + ibi_raw[5]) / 6
1246     ibi_e4_mean = (ibi_raw[6] + ibi_raw[7] + ibi_raw[8] + \
1247                  ibi_raw[9] + ibi_raw[10] + ibi_raw[11]) / 6
1248
1249     ibi_s3.append(ibi_s3_mean)
1250     ibi_e4.append(ibi_e4_mean)
1251

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1252     rmssd_raw = hp.get_data(data_path, column_name = 'Rmssd')
1253     rmssd_s3_mean = (rmssd_raw[0] + rmssd_raw[1] + rmssd_raw[2] + \
1254                     rmssd_raw[3] + rmssd_raw[4] + rmssd_raw[5]) / 6
1255     rmssd_e4_mean = (rmssd_raw[6] + rmssd_raw[7] + rmssd_raw[8] + \
1256                     rmssd_raw[9] + rmssd_raw[10] + rmssd_raw[11]) / 6
1257
1258     rmssd_s3.append(rmssd_s3_mean)
1259     rmssd_e4.append(rmssd_e4_mean)
1260
1261     sdnn_raw = hp.get_data(data_path, column_name = 'Sdnn')
1262     sdnn_s3_mean = (sdnn_raw[0] + sdnn_raw[1] + sdnn_raw[2] + \
1263                     sdnn_raw[3] + sdnn_raw[4] + sdnn_raw[5]) / 6
1264     sdnn_e4_mean = (sdnn_raw[6] + sdnn_raw[7] + sdnn_raw[8] + \
1265                     sdnn_raw[9] + sdnn_raw[10] + sdnn_raw[11]) / 6
1266
1267     sdnn_s3.append(sdnn_s3_mean)
1268     sdnn_e4.append(sdnn_e4_mean)
1269
1270     return bpm_e4, bpm_s3, ibi_e4, ibi_s3, rmssd_e4, rmssd_s3, sdnn_e4, sdnn_s3
1271
1272 def create_parameter_file(subjects, bpm_e4, bpm_s3, ibi_e4, ibi_s3, rmssd_e4, \
1273                           rmssd_s3, sdnn_e4, sdnn_s3):
1274     rows = zip(subjects, bpm_e4, bpm_s3, ibi_e4, ibi_s3, rmssd_e4, rmssd_s3, \
1275               sdnn_e4, sdnn_s3)
1276
1277     data_path = 'Summary_HR_Parameter.csv'
1278
1279     with open(data_path, "w") as f:
1280         writer = csv.writer(f)
1281         writer.writerow(['Subject', 'bpm_e4', 'bpm_s3', 'ibi_e4', 'ibi_s3', 'rmssd_e4', '
1282                           rmssd_s3', 'sdnn_e4', 'sdnn_s3'])
1283
1284         for row in rows:
1285             writer.writerow(row)
1286
1287     print('Successful FileCreation: ' + str(data_path))
1288
1289 #MAIN PART
1290 subj = 19
1291
1292 for subj in range(1, (subj + 1)):
1293     #IMPORT E4 DATA
1294     bvp_e4_adj = read_bvp_data(subj)
1295
1296     ibi_e4_adj, ibt_e4_adj = read_ibi_data(subj)
1297
1298     hr_e4_adj = read_hr_data(subj)
1299
1300     #CREATE TIME LISTS FOR E4
1301     time_e4_bvp = create_bvp_time(subj, bvp_e4_adj, timestamps_bvp)
1302
1303     time_e4_ibi = create_ibi_time(subj, ibt_e4_adj, timestamps_ibi)
1304
1305     time_e4_hr = create_hr_time(subj, hr_e4_adj, timestamps_hr)
1306
1307     #GET BVP LENGTH
1308     get_bvp_length(subj)
1309
1310
1311
1312     #IMPORT S3 DATA
1313     time_s3_adj, ppg_s3_adj, ibi_s3_adj, hr_s3_adj = read_s3_data(subj, time_s3_adj,

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1314     ppg_s3_adj, ibi_s3_adj, hr_s3_adj)
1315
1316     #CUT RECORDINGS IN SHAPE
1317     tppg_s3_adj, tibi_s3_adj, thr_s3_adj, ppg_s3_adj, ibi_s3_adj, hr_s3_adj, bvp_e4_adj,
1318     ibi_e4_adj, hr_e4_adj, time_e4_bvp, time_e4_ibi, time_e4_hr = cut_in_shape(subj,
1319     time_s3_adj, ppg_s3_adj, ibi_s3_adj, hr_s3_adj, bvp_e4_adj, ibi_e4_adj, hr_e4_adj,
1320     time_e4_bvp, time_e4_ibi, time_e4_hr)
1321
1322     #SPLIT UP E4 DATA
1323     bvp_e4_bs1, bvp_e4_st1, bvp_e4_rc1, bvp_e4_bs2, bvp_e4_st2, bvp_e4_rc2, tbvp_e4_bs1,
1324     tbvp_e4_st1, tbvp_e4_rc1, tbvp_e4_bs2, tbvp_e4_st2, tbvp_e4_rc2 = split_up_e4_bvp(subj,
1325     bvp_e4_adj, time_e4_bvp, bvp_e4_bs1, bvp_e4_st1, bvp_e4_rc1, bvp_e4_bs2, bvp_e4_st2,
1326     bvp_e4_rc2, tbvp_e4_bs1, tbvp_e4_st1, tbvp_e4_rc1, tbvp_e4_bs2, tbvp_e4_st2, tbvp_e4_rc2
1327     )
1328     ibi_e4_bs1, ibi_e4_st1, ibi_e4_rc1, ibi_e4_bs2, ibi_e4_st2, ibi_e4_rc2, tibi_e4_bs1,
1329     tibi_e4_st1, tibi_e4_rc1, tibi_e4_bs2, tibi_e4_st2, tibi_e4_rc2 = split_up_e4_ibi(subj,
1330     ibi_e4_adj, time_e4_ibi, ibi_e4_bs1, ibi_e4_st1, ibi_e4_rc1, ibi_e4_bs2, ibi_e4_st2,
1331     ibi_e4_rc2, tibi_e4_bs1, tibi_e4_st1, tibi_e4_rc1, tibi_e4_bs2, tibi_e4_st2, tibi_e4_rc2
1332     )
1333     hr_e4_bs1, hr_e4_st1, hr_e4_rc1, hr_e4_bs2, hr_e4_st2, hr_e4_rc2, thr_e4_bs1, thr_e4_st1,
1334     thr_e4_rc1, thr_e4_bs2, thr_e4_st2, thr_e4_rc2 = split_up_e4_hr(subj, hr_e4_adj,
1335     time_e4_hr, hr_e4_bs1, hr_e4_st1, hr_e4_rc1, hr_e4_bs2, hr_e4_st2, hr_e4_rc2, thr_e4_bs1,
1336     thr_e4_st1, thr_e4_rc1, thr_e4_bs2, thr_e4_st2, thr_e4_rc2)
1337
1338     #SPLIT UP S3 DATA
1339     ppg_s3_bs1, ppg_s3_st1, ppg_s3_rc1, ppg_s3_bs2, ppg_s3_st2, ppg_s3_rc2, tppg_s3_bs1,
1340     tppg_s3_st1, tppg_s3_rc1, tppg_s3_bs2, tppg_s3_st2, tppg_s3_rc2 = split_up_s3_ppg(subj,
1341     tppg_s3_adj, ppg_s3_adj, ppg_s3_bs1, ppg_s3_st1, ppg_s3_rc1, ppg_s3_bs2, ppg_s3_st2,
1342     ppg_s3_rc2, tppg_s3_bs1, tppg_s3_st1, tppg_s3_rc1, tppg_s3_bs2, tppg_s3_st2, tppg_s3_rc2
1343     )
1344     ibi_s3_bs1, ibi_s3_st1, ibi_s3_rc1, ibi_s3_bs2, ibi_s3_st2, ibi_s3_rc2, tibi_s3_bs1,
1345     tibi_s3_st1, tibi_s3_rc1, tibi_s3_bs2, tibi_s3_st2, tibi_s3_rc2 = split_up_s3_ibi(subj,
1346     tibi_s3_adj, ibi_s3_adj, ibi_s3_bs1, ibi_s3_st1, ibi_s3_rc1, ibi_s3_bs2, ibi_s3_st2,
1347     ibi_s3_rc2, tibi_s3_bs1, tibi_s3_st1, tibi_s3_rc1, tibi_s3_bs2, tibi_s3_st2, tibi_s3_rc2
1348     )
1349     hr_s3_bs1, hr_s3_st1, hr_s3_rc1, hr_s3_bs2, hr_s3_st2, hr_s3_rc2, thr_s3_bs1, thr_s3_st1,
1350     thr_s3_rc1, thr_s3_bs2, thr_s3_st2, thr_s3_rc2 = split_up_s3_hr(subj, thr_s3_adj,
1351     hr_s3_adj, hr_s3_bs1, hr_s3_st1, hr_s3_rc1, hr_s3_bs2, hr_s3_st2, hr_s3_rc2, thr_s3_bs1,
1352     thr_s3_st1, thr_s3_rc1, thr_s3_bs2, thr_s3_st2, thr_s3_rc2)
1353
1354     #DOWNSAMPLE S3 DATA
1355     ppg_s3_bs1, ppg_s3_st1, ppg_s3_rc1, ppg_s3_bs2, ppg_s3_st2, ppg_s3_rc2 =
1356     downsampling_ppg(subj, tppg_s3_bs1, tppg_s3_st1, tppg_s3_rc1, tppg_s3_bs2, tppg_s3_st2,
1357     tppg_s3_rc2, bvp_e4_bs1, bvp_e4_st1, bvp_e4_rc1, bvp_e4_bs2, bvp_e4_st2, bvp_e4_rc2,
1358     ppg_s3_bs1, ppg_s3_st1, ppg_s3_rc1, ppg_s3_bs2, ppg_s3_st2, ppg_s3_rc2)
1359     ibi_s3_bs1, ibi_s3_st1, ibi_s3_rc1, ibi_s3_bs2, ibi_s3_st2, ibi_s3_rc2, zero_ibi_bs1,
1360     zero_ibi_st1, zero_ibi_rc1, zero_ibi_bs2, zero_ibi_st2, zero_ibi_rc2 = downsampling_ibi(
1361     subj, zero_ibi_bs1, zero_ibi_st1, zero_ibi_rc1, zero_ibi_bs2, zero_ibi_st2, zero_ibi_rc2,
1362     tibi_s3_bs1, tibi_s3_st1, tibi_s3_rc1, tibi_s3_bs2, tibi_s3_st2, tibi_s3_rc2,
1363     ibi_e4_bs1, ibi_e4_st1, ibi_e4_rc1, ibi_e4_bs2, ibi_e4_st2, ibi_e4_rc2, ibi_s3_bs1,
1364     ibi_s3_st1, ibi_s3_rc1, ibi_s3_bs2, ibi_s3_st2, ibi_s3_rc2)
1365     hr_s3_bs1, hr_s3_st1, hr_s3_rc1, hr_s3_bs2, hr_s3_st2, hr_s3_rc2, zero_hr_bs1,
1366     zero_hr_st1, zero_hr_rc1, zero_hr_bs2, zero_hr_st2, zero_hr_rc2 = downsampling_hr(subj,

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

zero_hr_bs1, zero_hr_st1, zero_hr_rc1, zero_hr_bs2, zero_hr_st2, zero_hr_rc2, thr_s3_bs1
, thr_s3_st1, thr_s3_rc1, thr_s3_bs2, thr_s3_st2, thr_s3_rc2, hr_e4_bs1, hr_e4_st1,
hr_e4_rc1, hr_e4_bs2, hr_e4_st2, hr_e4_rc2, hr_s3_bs1, hr_s3_st1, hr_s3_rc1, hr_s3_bs2,
hr_s3_st2, hr_s3_rc2)

1342
1343 #bvp_e4_bs1, bvp_e4_st1, bvp_e4_rc1, bvp_e4_bs2, bvp_e4_st2, bvp_e4_rc2, tbvp_e4_bs1,
tbvp_e4_st1, tbvp_e4_rc1, tbvp_e4_bs2, tbvp_e4_st2, tbvp_e4_rc2 = upsampling_bvp(subj,
ppg_s3_bs1, ppg_s3_st1, ppg_s3_rc1, ppg_s3_bs2, ppg_s3_st2, ppg_s3_rc2, bvp_e4_bs1,
bvp_e4_st1, bvp_e4_rc1, bvp_e4_bs2, bvp_e4_st2, bvp_e4_rc2, tbvp_e4_bs1, tbvp_e4_st1,
tbvp_e4_rc1, tbvp_e4_bs2, tbvp_e4_st2, tbvp_e4_rc2)

1344
1345
1346 #HEARTPY
1347 if subj == 3 or subj == 16:
1348     print('Bad Quality')
1349 else:
1350     get_hr_s3_parameter(subj, 'PPG_S3_BS1', ppg_s3_bs1)
1351     get_hr_s3_parameter(subj, 'PPG_S3_ST1', ppg_s3_st1)
1352     get_hr_s3_parameter(subj, 'PPG_S3_RC1', ppg_s3_rc1)
1353     get_hr_s3_parameter(subj, 'PPG_S3_BS2', ppg_s3_bs2)
1354     get_hr_s3_parameter(subj, 'PPG_S3_ST2', ppg_s3_st2)
1355     get_hr_s3_parameter(subj, 'PPG_S3_RC2', ppg_s3_rc2)
1356
1357     get_hr_e4_parameter(subj, 'PPG_E4_BS1', bvp_e4_bs1, tbvp_e4_bs1)
1358     get_hr_e4_parameter(subj, 'PPG_E4_ST1', bvp_e4_st1, tbvp_e4_st1)
1359     get_hr_e4_parameter(subj, 'PPG_E4_RC1', bvp_e4_rc1, tbvp_e4_rc1)
1360     get_hr_e4_parameter(subj, 'PPG_E4_BS2', bvp_e4_bs2, tbvp_e4_bs2)
1361     get_hr_e4_parameter(subj, 'PPG_E4_ST2', bvp_e4_st2, tbvp_e4_st2)
1362     get_hr_e4_parameter(subj, 'PPG_E4_RC2', bvp_e4_rc2, tbvp_e4_rc2)
1363
1364     save_hr_parameters(subj, modes, bpm, breathingrate, hr_mad, ibi, pnn20, pnn50, rmssd
, s, sd1, sd1_sd2, sd2, sdn, sdsd)

1365
1366 #CREATE FILES
1367 create_file(subj, 'PPG_S3_BS1', ppg_s3_bs1[0], ppg_s3_bs1[1])
1368 create_file(subj, 'PPG_S3_ST1', ppg_s3_st1[0], ppg_s3_st1[1])
1369 create_file(subj, 'PPG_S3_RC1', ppg_s3_rc1[0], ppg_s3_rc1[1])
1370 create_file(subj, 'PPG_S3_BS2', ppg_s3_bs2[0], ppg_s3_bs2[1])
1371 create_file(subj, 'PPG_S3_ST2', ppg_s3_st2[0], ppg_s3_st2[1])
1372 create_file(subj, 'PPG_S3_RC2', ppg_s3_rc2[0], ppg_s3_rc2[1])
1373
1374 create_file(subj, 'PPG_E4_BS1', bvp_e4_bs1, tbvp_e4_bs1)
1375 create_file(subj, 'PPG_E4_ST1', bvp_e4_st1, tbvp_e4_st1)
1376 create_file(subj, 'PPG_E4_RC1', bvp_e4_rc1, tbvp_e4_rc1)
1377 create_file(subj, 'PPG_E4_BS2', bvp_e4_bs2, tbvp_e4_bs2)
1378 create_file(subj, 'PPG_E4_ST2', bvp_e4_st2, tbvp_e4_st2)
1379 create_file(subj, 'PPG_E4_RC2', bvp_e4_rc2, tbvp_e4_rc2)
1380
1381 if zero_ibi_bs1 == False:
1382     create_file(subj, 'IBI_S3_BS1', ibi_s3_bs1[0], ibi_s3_bs1[1])
1383     create_file(subj, 'IBI_E4_BS1', ibi_e4_bs1, tibi_e4_bs1)
1384
1385 if zero_ibi_st1 == False:
1386     create_file(subj, 'IBI_S3_ST1', ibi_s3_st1[0], ibi_s3_st1[1])
1387     create_file(subj, 'IBI_E4_ST1', ibi_e4_st1, tibi_e4_st1)
1388
1389 if zero_ibi_rc1 == False:
1390     create_file(subj, 'IBI_S3_RC1', ibi_s3_rc1[0], ibi_s3_rc1[1])
1391     create_file(subj, 'IBI_E4_RC1', ibi_e4_rc1, tibi_e4_rc1)
1392
1393 if zero_ibi_bs2 == False:
1394     create_file(subj, 'IBI_S3_BS2', ibi_s3_bs2[0], ibi_s3_bs2[1])
1395     create_file(subj, 'IBI_E4_BS2', ibi_e4_bs2, tibi_e4_bs2)

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1396
1397     if zero_ibi_st2 == False:
1398         create_file(subj, 'IBI_S3_ST2', ibi_s3_st2[0], ibi_s3_st2[1])
1399         create_file(subj, 'IBI_E4_ST2', ibi_e4_st2, tibi_e4_st2)
1400
1401     if zero_ibi_rc2 == False:
1402         create_file(subj, 'IBI_S3_RC2', ibi_s3_rc2[0], ibi_s3_rc2[1])
1403         create_file(subj, 'IBI_E4_RC2', ibi_e4_rc2, tibi_e4_rc2)
1404
1405
1406     if zero_hr_bs1 == False:
1407         create_file(subj, 'HR_S3_BS1', hr_s3_bs1[0], hr_s3_bs1[1])
1408         create_file(subj, 'HR_E4_BS1', hr_e4_bs1, thr_e4_bs1)
1409
1410     if zero_hr_st1 == False:
1411         create_file(subj, 'HR_S3_ST1', hr_s3_st1[0], hr_s3_st1[1])
1412         create_file(subj, 'HR_E4_ST1', hr_e4_st1, thr_e4_st1)
1413
1414     if zero_hr_rc1 == False:
1415         create_file(subj, 'HR_S3_RC1', hr_s3_rc1[0], hr_s3_rc1[1])
1416         create_file(subj, 'HR_E4_RC1', hr_e4_rc1, thr_e4_rc1)
1417
1418     if zero_hr_bs2 == False:
1419         create_file(subj, 'HR_S3_BS2', hr_s3_bs2[0], hr_s3_bs2[1])
1420         create_file(subj, 'HR_E4_BS2', hr_e4_bs2, thr_e4_bs2)
1421
1422     if zero_hr_st2 == False:
1423         create_file(subj, 'HR_S3_ST2', hr_s3_st2[0], hr_s3_st2[1])
1424         create_file(subj, 'HR_E4_ST2', hr_e4_st2, thr_e4_st2)
1425
1426     if zero_hr_rc2 == False:
1427         create_file(subj, 'HR_S3_RC2', hr_s3_rc2[0], hr_s3_rc2[1])
1428         create_file(subj, 'HR_E4_RC2', hr_e4_rc2, thr_e4_rc2)
1429
1430
1431     #CROSSCORRELATIONS
1432     np.cross_correlation(subj, 'PPG_BS1', ppg_s3_bs1[0], bvp_e4_bs1)
1433     np.cross_correlation(subj, 'PPG_ST1', ppg_s3_st1[0], bvp_e4_st1)
1434     np.cross_correlation(subj, 'PPG_RC1', ppg_s3_rc1[0], bvp_e4_rc1)
1435     np.cross_correlation(subj, 'PPG_BS2', ppg_s3_bs2[0], bvp_e4_bs2)
1436     np.cross_correlation(subj, 'PPG_ST2', ppg_s3_st2[0], bvp_e4_st2)
1437     np.cross_correlation(subj, 'PPG_RC2', ppg_s3_rc2[0], bvp_e4_rc2)
1438
1439     if zero_ibi_bs1 == False:
1440         np.cross_correlation(subj, 'IBI_BS1', ibi_s3_bs1[0], ibi_e4_bs1)
1441     if zero_ibi_st1 == False:
1442         np.cross_correlation(subj, 'IBI_ST1', ibi_s3_st1[0], ibi_e4_st1)
1443     if zero_ibi_rc1 == False:
1444         np.cross_correlation(subj, 'IBI_RC1', ibi_s3_rc1[0], ibi_e4_rc1)
1445     if zero_ibi_bs2 == False:
1446         np.cross_correlation(subj, 'IBI_BS2', ibi_s3_bs2[0], ibi_e4_bs2)
1447     if zero_ibi_st2 == False:
1448         np.cross_correlation(subj, 'IBI_ST2', ibi_s3_st2[0], ibi_e4_st2)
1449     if zero_ibi_rc2 == False:
1450         np.cross_correlation(subj, 'IBI_RC2', ibi_s3_rc2[0], ibi_e4_rc2)
1451
1452     if zero_ibi_bs1 == False:
1453         np.cross_correlation(subj, 'HR_BS1', hr_s3_bs1[0], hr_e4_bs1)
1454     if zero_ibi_st1 == False:
1455         np.cross_correlation(subj, 'HR_ST1', hr_s3_st1[0], hr_e4_st1)
1456     if zero_ibi_rc1 == False:
1457         np.cross_correlation(subj, 'HR_RC1', hr_s3_rc1[0], hr_e4_rc1)
1458     if zero_ibi_bs2 == False:

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1459     np_cross_correlation(subj, 'HR_BS2', hr_s3_bs2[0], hr_e4_bs2)
1460 if zero_ibi_st2 == False:
1461     np_cross_correlation(subj, 'HR_ST2', hr_s3_st2[0], hr_e4_st2)
1462 if zero_ibi_rc2 == False:
1463     np_cross_correlation(subj, 'HR_RC2', hr_s3_rc2[0], hr_e4_rc2)
1464
1465 print(corr_coef)
1466 print(corr_mode)
1467
1468 if subj == 19:
1469     create_corr_file('CorrCoef', corr_coef, corr_mode)
1470
1471 #PARAMETER TRANSFORMATION
1472 if subj == 3 or subj == 16:
1473     print('No Files available')
1474 else:
1475     read_hr_parameters(subj, bpm_e4, bpm_s3, ibi_e4, ibi_s3, rmssd_e4, rmssd_s3, sdnne4,
1476                        sdnne_s3)
1477
1478 #CLEAN LISTS
1479 time_s3_adj = []
1480 ppg_s3_adj = []
1481 ibi_s3_adj = []
1482 hr_s3_adj = []
1483
1484 bvp_e4_adj = []
1485 ibi_e4_adj = []
1486 ibt_e4_adj = []
1487 hr_e4_adj = []
1488
1489 ppg_s3_bs1 = []
1490 ppg_s3_st1 = []
1491 ppg_s3_rc1 = []
1492 ppg_s3_bs2 = []
1493 ppg_s3_st2 = []
1494 ppg_s3_rc2 = []
1495
1496 ibi_s3_bs1 = []
1497 ibi_s3_st1 = []
1498 ibi_s3_rc1 = []
1499 ibi_s3_bs2 = []
1500 ibi_s3_st2 = []
1501 ibi_s3_rc2 = []
1502
1503 hr_s3_bs1 = []
1504 hr_s3_st1 = []
1505 hr_s3_rc1 = []
1506 hr_s3_bs2 = []
1507 hr_s3_st2 = []
1508 hr_s3_rc2 = []
1509
1510 bvp_e4_bs1 = []
1511 bvp_e4_st1 = []
1512 bvp_e4_rc1 = []
1513 bvp_e4_bs2 = []
1514 bvp_e4_st2 = []
1515 bvp_e4_rc2 = []
1516
1517 ibi_e4_bs1 = []
1518 ibi_e4_st1 = []
1519 ibi_e4_rc1 = []
1520 ibi_e4_bs2 = []
1521 ibi_e4_st2 = []

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1521     ibi_e4_rc2 = []
1522
1523     hr_e4_bs1 = []
1524     hr_e4_st1 = []
1525     hr_e4_rc1 = []
1526     hr_e4_bs2 = []
1527     hr_e4_st2 = []
1528     hr_e4_rc2 = []
1529
1530     tppg_s3_bs1 = []
1531     tppg_s3_st1 = []
1532     tppg_s3_rc1 = []
1533     tppg_s3_bs2 = []
1534     tppg_s3_st2 = []
1535     tppg_s3_rc2 = []
1536
1537     tibi_s3_bs1 = []
1538     tibi_s3_st1 = []
1539     tibi_s3_rc1 = []
1540     tibi_s3_bs2 = []
1541     tibi_s3_st2 = []
1542     tibi_s3_rc2 = []
1543
1544     thr_s3_bs1 = []
1545     thr_s3_st1 = []
1546     thr_s3_rc1 = []
1547     thr_s3_bs2 = []
1548     thr_s3_st2 = []
1549     thr_s3_rc2 = []
1550
1551     tbvp_e4_bs1 = []
1552     tbvp_e4_st1 = []
1553     tbvp_e4_rc1 = []
1554     tbvp_e4_bs2 = []
1555     tbvp_e4_st2 = []
1556     tbvp_e4_rc2 = []
1557
1558     tibi_e4_bs1 = []
1559     tibi_e4_st1 = []
1560     tibi_e4_rc1 = []
1561     tibi_e4_bs2 = []
1562     tibi_e4_st2 = []
1563     tibi_e4_rc2 = []
1564
1565     thr_e4_bs1 = []
1566     thr_e4_st1 = []
1567     thr_e4_rc1 = []
1568     thr_e4_bs2 = []
1569     thr_e4_st2 = []
1570     thr_e4_rc2 = []
1571
1572     zero_ibi_bs1 = False
1573     zero_ibi_st1 = False
1574     zero_ibi_rc1 = False
1575     zero_ibi_bs2 = False
1576     zero_ibi_st2 = False
1577     zero_ibi_rc2 = False
1578
1579     zero_hr_bs1 = False
1580     zero_hr_st1 = False
1581     zero_hr_rc1 = False
1582     zero_hr_bs2 = False
1583     zero_hr_st2 = False

```

VALIDATION OF A WRIST-WORN PPG SENSOR

```

1584     zero_hr_rc2 = False
1585
1586     modes          = [ 'Modes' ]
1587     bpm            = [ 'Bpm' ]
1588     breathingrate  = [ 'Breathingrate' ]
1589     hf            = [ 'Hf' ]
1590     hr_mad        = [ 'Hr_mad' ]
1591     ibi           = [ 'Ibi' ]
1592     lf            = [ 'Lf' ]
1593     lf_hf         = [ 'Lf/Hf' ]
1594     pnn20         = [ 'Pnn20' ]
1595     pnn50         = [ 'Pnn50' ]
1596     rmssd         = [ 'Rmssd' ]
1597     s             = [ 'S' ]
1598     sd1           = [ 'Sd1' ]
1599     sd1_sd2       = [ 'Sd1/Sd2' ]
1600     sd2           = [ 'Sd2' ]
1601     sdnns         = [ 'Sdnns' ]
1602     sdsd          = [ 'Sdsd' ]
1603
1604     print( 'ALL LISTS ARE CLEANED!' )
1605
1606     create_parameter_file( subjects , bpm_e4 , bpm_s3 , ibi_e4 , ibi_s3 , rmssd_e4 , rmssd_s3 , sdnns_e4 ,
                           sdnns_s3 )

```


VALIDATION OF A WRIST-WORN PPG SENSOR

Appendix F – Bland-Altman Plot (R-Script)

```

1  install.packages("blandr")
2  library(blandr)
3  summary <- read.csv("Summary_HR_Parameter.csv", header=TRUE, sep = ",", quote = "\"", dec =
    ".", fill = TRUE)
4
5  statistics.results_hr <- blandr.statistics(summary$bpm_e4 , summary$bpm_s3 )
6
7  jpeg("Bland-Altman-Plot_HR_neu.jpeg", width = 1000, height = 500)
8  blandr.plot.ggplot(statistics.results_hr, method1name = "E4",
9                    method2name = "S3",
10                   ciDisplay = TRUE, ciShading = TRUE, normalLow = FALSE,
11                   normalHigh = FALSE, overlapping = FALSE,
12                   plotProportionalBias = FALSE,
13                   plotProportionalBias.se = TRUE, assume.differences.are.normal = TRUE)+
14    geom_hline(yintercept=5, size=1.1)+
15    geom_hline(yintercept=-5, size=1.1)+
16    geom_point(shape=16, size=4)+
17    labs(title = "HR", x="Mean HR (E4,S3)", y = "Difference HR (E4-S3)")+
18    theme(plot.title = element_text(size = 20, face = "bold"), axis.title.x=element_text(size
    =15), axis.title.y=element_text(size=15))
19  dev.off()
20
21  statistics.results_sdnn <- blandr.statistics(summary$sdnn_e4 , summary$sdnn_s3 )
22
23  jpeg("Bland-Altman-Plot_SDNN_neu.jpeg", width = 1000, height = 500)
24  blandr.plot.ggplot(statistics.results_sdnn, method1name = "E4",
25                    method2name = "S3",
26                   ciDisplay = TRUE, ciShading = TRUE, normalLow = FALSE,
27                   normalHigh = FALSE, overlapping = FALSE,
28                   plotProportionalBias = FALSE,
29                   plotProportionalBias.se = TRUE, assume.differences.are.normal = TRUE)+
30    geom_hline(yintercept=0.56, size=1.1)+
31    geom_hline(yintercept=-0.56, size=1.1)+
32    geom_point(shape=16, size=4)+
33    labs(title = "SDNN", x="Mean SDNN (E4,S3)", y = "Difference SDNN (E4-S3)")+
34    theme(plot.title = element_text(size = 20, face = "bold"), axis.title.x=element_text(size
    =15), axis.title.y=element_text(size=15))
35  dev.off()
36
37  statistics.results_rmssd <- blandr.statistics(summary$rmssd_e4 , summary$rmssd_s3 )
38
39  jpeg("Bland-Altman-Plot_RMSSD_neu.jpeg", width = 1000, height = 500)
40  blandr.plot.ggplot(statistics.results_rmssd, method1name = "E4",
41                    method2name = "S3",
42                   ciDisplay = TRUE, ciShading = TRUE, normalLow = FALSE,
43                   normalHigh = FALSE, overlapping = FALSE,
44                   plotProportionalBias = FALSE,
45                   plotProportionalBias.se = TRUE, assume.differences.are.normal = TRUE)+
46    geom_hline(yintercept=0.71, size=1.1)+
47    geom_hline(yintercept=-0.71, size=1.1)+
48    geom_point(shape=16, size=4)+
49    labs(title = "RMSSD", x="Mean RMSSD (E4,S3)", y = "Difference RMSSD (E4-S3)")+
50    theme(plot.title = element_text(size = 20, face = "bold"), axis.title.x=element_text(size
    =15), axis.title.y=element_text(size=15))
51  dev.off()

```

VALIDATION OF A WRIST-WORN PPG SENSOR

Appendix G – Error Bar Plot (R-Script)

```

1 rm(list = ls())
2 #####Task1
3 library(ggplot2)
4 library(grDevices)
5 task1 <- read.csv("Summary_1st_Task.csv", header=TRUE, sep = ",", quote = "\"'", dec = ".",
6                 fill = TRUE)
7
8 zero<- c(0,0,0)
9 boun1<- c(6.809,6.809,6.809)
10 boun2<- c(-6.809,-6.809,-6.809)
11 data<- cbind(task1, zero, boun1, boun2)
12 windows(width=10, height=5)
13 ggplot(data, aes(x = factor(Phase, level = c('BS1', 'ST1', 'RC1')), group=1)) +
14   geom_line(aes(y=p1), size = 1, color= "darkgrey") +
15   geom_line(aes(y=p2), size = 1, color= "darkgrey")+
16   geom_line(aes(y=p4), size = 1, color= "darkgrey")+
17   geom_line(aes(y=p5), size = 1, color= "darkgrey")+
18   geom_line(aes(y=p6), size = 1, color= "darkgrey")+
19   geom_line(aes(y=p7), size = 1, color= "darkgrey")+
20   geom_line(aes(y=p9), size = 1, color= "darkgrey")+
21   geom_line(aes(y=p15), size = 1, color= "darkgrey")+
22   geom_line(aes(y=p18), size = 1, color= "darkgrey")+
23   geom_line(aes(y=zero), size = 1, color= "black")+
24   geom_line(aes(y=boun1), size = 1, color= "green")+
25   geom_line(aes(y=boun2), size = 1, color= "green")+
26   geom_line(aes(y=mean), size= 1, color = "red")+
27   geom_errorbar(aes(ymin = mean-se, ymax= mean + se),color="red", size=1, width=.2)+
28   labs(title = "Modified SSST (E4-S3)", x="", y = "Heart Rate: E4 - S3 (bpm)")
29 #####Task2
30 rm(list = ls())
31 task2 <- read.csv("Summary_2nd_Task.csv", header=TRUE, sep = ",", quote = "\"'", dec = ".",
32                 fill = TRUE)
33
34 zero<- c(0,0,0)
35 boun1<- c(13.33,13.33,13.33)
36 boun2<- c(-13.33,-13.33,-13.33)
37 data<- cbind(task2, zero, boun1, boun2)
38 windows(width=10, height=5)
39 ggplot(data, aes(x = factor(Phase, level = c('BS2', 'ST2', 'RC2')), group=1)) +
40   geom_line(aes(y=p2), size = 1, color= "darkgrey")+
41   geom_line(aes(y=p5), size = 1, color= "darkgrey")+
42   geom_line(aes(y=p7), size = 1, color= "darkgrey")+
43   geom_line(aes(y=p15), size = 1, color= "darkgrey")+
44   geom_line(aes(y=p18), size = 1, color= "darkgrey")+
45   geom_line(aes(y=zero), size = 1, color= "black")+
46   geom_line(aes(y=boun1), size = 1, color= "green")+
47   geom_line(aes(y=boun2), size = 1, color= "green")+
48   geom_line(aes(y=mean), size= 1, color = "red")+
49   geom_errorbar(aes(ymin = mean-se, ymax= mean + se),color="red", size=1, width=.2)+
50   labs(title = "Stroop-Task (E4-S3)", x="", y = "Heart Rate: E4 - S3 (bpm)")

```

VALIDATION OF A WRIST-WORN PPG SENSOR

Appendix H – Overview of mean STAI scores of participants per phase

Participant	T1 (before SSST)	T2 (after SSST)	T3 (after Recovery)	T4 (before Stroop)	T5 (after Stroop)	T6 (after Recovery)
1	40	53.33	23.33	26.66	40	20
2	23.33	23.33	23.33	23.33	30	26.66
3	46.66	53.33	50	43.33	46.66	40
4	33.33	56.66	26.66	36.66	46.66	30
5	30	43.33	33.33	40	30	36.66
6	30	26.66	23.33	20	43.33	30
7	43.33	43.33	33.33	26.66	43.33	33.33
8	40	43.33	30	43.33	50	43.33
9	53.33	43.33	46.66	50	56.66	46.66
10	46.66	63.33	40	20	30	40
11	40	40	23.33	20	36.66	23.33
12	36.66	36.66	33.33	33.33	33.33	33.33
13	66.66	56.66	36.66	46.66	60	43.33
14	30	43.33	43.33	30	40	30
15	26.66	33.33	30	33.33	40	33.33
16	43.33	46.66	53.33	23.33	50	26.66
17	33.33	46.66	26.66	23.33	30	23.33
18	33.33	26.66	26.66	20	33.33	20
19	36.66	46.66	30	40	36.66	46.66
20	40	43.33	36.66	33.33	40	30