



UNIVERSITÀ DEGLI STUDI  
DI TRENTO

---

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE  
ICT International Doctoral School

A DATA ANALYTICS FRAMEWORK FOR  
PHYSIOLOGICAL SIGNALS FROM  
WEARABLE DEVICES

Andrea Bizzego

Advisor

Dr. Cesare Furlanello

Università degli Studi di Trento

---

January 2017



# Disclaimer

The work presented in this PhD thesis has been funded by a scholarship from *TIM - Telecom Italia*: results might be covered by Intellectual Property Rights and Patents.



# Abstract

*Wearable devices have emerged as the most innovative opportunity to enable acquisition and quantification of physiological signals in real-world indoor or outdoor contexts. However, their use in research should be based on a reproducible analytics process, ensuring that all the critical steps in data collection and processing are managed in a reliable experimental setup. The aim of this thesis is to investigate the actual value and technical limitations of wearable devices for their use in a research context, such as physiological monitoring of sleep and crying states in infants, of parenting of typical or atypical children, synchrony in educational contexts, and of fatigue patterns in outdoor sport activity, e.g. skiing. The thesis describes an approach and solutions that aim to compensate the effects of such technical limits. Besides providing a set of appropriate signal processing algorithms, a real-life sensing architecture is designed and implemented enabling synchronized acquisition from multiple subjects and multiple sensors, including cardiac signals, electrodermal activity and inertial data streams. The signal processing pipeline and the real-life sensing architecture are merged in a unique data analytics framework (Physiolitix). The framework is validated on a fairly wide range of sensors, including medical quality multi-sensor smartwatches and smart textile garments applied in diverse research contexts. In particular, a calibration dataset is developed to compare wearable and clinical devices in an affective computing task. We found that wearables can be employed as a valid substitute for medical quality devices with the help of adequate signal processing and machine learning solutions.*

## **Keywords**

wearable devices, physiological signal processing, affective computing, real-life sensing



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Physiolitix</b>	<b>5</b>
2.1	Wearable Devices for research . . . . .	6
2.1.1	Physiological signals and sensors . . . . .	8
2.1.2	Technical aspects . . . . .	13
2.1.3	Issues intrinsic to WD usage . . . . .	17
2.1.4	Existing platforms . . . . .	20
2.2	Result: Physiolitix . . . . .	21
2.2.1	PhysioREC: real-life sensing . . . . .	22
2.2.2	PhysioBackend . . . . .	28
2.3	Discussion . . . . .	30
<b>3</b>	<b>pyPhysio</b>	<b>33</b>
3.1	Physiological Signal Processing . . . . .	34
3.2	Structure of pyPhysio . . . . .	36
3.2.1	class <code>Signal</code> . . . . .	36
3.2.2	class <code>Algorithm</code> . . . . .	38
3.3	Physiological signals . . . . .	39
3.3.1	Cardiac signals . . . . .	41
3.3.2	Electrodermal Activity . . . . .	41
3.3.3	Activity signals . . . . .	41

3.4	Discussion . . . . .	42
<b>4</b>	<b>WCS dataset</b>	<b>45</b>
4.1	Devices and architecture . . . . .	46
4.1.1	Selection of WDs . . . . .	46
4.1.2	The baseline: Thought Technology FlexComp . . . . .	48
4.1.3	Empatica E4 . . . . .	49
4.1.4	ComfTech HeartBand . . . . .	50
4.1.5	PhysioREC . . . . .	51
4.2	Synchronization . . . . .	52
4.3	Experimental phases . . . . .	53
4.3.1	Baseline and moving . . . . .	53
4.3.2	Stimulation . . . . .	54
4.4	Results . . . . .	56
4.5	Discussion . . . . .	57
<b>5</b>	<b>Reproducibility</b>	<b>59</b>
5.1	Modeling and validation . . . . .	62
5.1.1	Definition of problem . . . . .	63
5.1.2	Signal processing pipelines . . . . .	63
5.1.3	Machine Learning pipeline . . . . .	65
5.1.4	Classification performances . . . . .	67
5.2	Results . . . . .	69
5.2.1	Comparison with background studies . . . . .	69
5.2.2	Performances on WCS-Medical . . . . .	72
5.2.3	Performances on WCS-WDs . . . . .	72
5.3	Discussion . . . . .	73
<b>6</b>	<b>Case studies</b>	<b>77</b>
6.1	HRV to detect infant states . . . . .	78

6.1.1	Aim of the study . . . . .	78
6.1.2	Materials and methods . . . . .	79
6.1.3	Results . . . . .	80
6.1.4	Conclusions . . . . .	81
6.2	Fatigue in skiers . . . . .	82
6.2.1	Aim of the study . . . . .	82
6.2.2	Materials and methods . . . . .	83
6.2.3	Results . . . . .	86
6.2.4	Conclusions . . . . .	87
6.3	Response to infant crying . . . . .	89
6.3.1	Aim of the study . . . . .	90
6.3.2	Materials and methods . . . . .	90
6.3.3	Results . . . . .	91
6.3.4	Conclusions . . . . .	92
6.4	Synchrony in music-therapy . . . . .	93
6.4.1	Materials and methods . . . . .	93
6.4.2	Initial and expected results . . . . .	94
<b>7</b>	<b>Conclusions</b>	<b>97</b>
	<b>Bibliography</b>	<b>101</b>
<b>A</b>	<b>DBD Algorithm</b>	<b>121</b>
A.1	Beat detection pipeline . . . . .	124
A.1.1	Oversampling . . . . .	124
A.1.2	Energy estimation . . . . .	124
A.1.3	The DBD algorithm . . . . .	126
A.1.4	The RCO algorithm . . . . .	127
A.2	Materials and methods . . . . .	130
A.2.1	Metrics . . . . .	131

A.3	Results . . . . .	132
A.3.1	Results on Fantasia dataset . . . . .	133
A.3.2	Results on WCS dataset . . . . .	133
A.4	Discussion . . . . .	136
<b>B</b>	<b>Phasic estimation</b>	<b>139</b>
B.0.1	Notation . . . . .	141
B.1	Loss function . . . . .	142
B.1.1	Stage 1: Identification of peaks . . . . .	144
B.1.2	Stage 2: Detrending . . . . .	145
B.1.3	Stage 3: Computation of energy . . . . .	146
B.2	Estimation of PSR component . . . . .	146
B.2.1	Stage 1: Identification of candidate peaks . . . . .	147
B.2.2	Stage 2: Identification of onset and termination in- stants . . . . .	147
B.2.3	Stage 3: Interpolation . . . . .	147
B.3	Materials and Methods . . . . .	147
B.3.1	Metrics . . . . .	148
B.3.2	Processing procedure . . . . .	148
B.4	Results . . . . .	150
B.5	Discussion . . . . .	152
<b>C</b>	<b>Wearable Devices</b>	<b>155</b>
C.1	Empatica E4 . . . . .	155
C.2	Mio Alpha 2/Link . . . . .	156
C.3	ComfTech smart garments . . . . .	157
C.4	Interaxon Muse . . . . .	157
C.5	Emotiv Epoc/+ . . . . .	158
C.6	EXEL Exls3 . . . . .	158

<b>D</b>	<b>pyPhysio tutorial</b>	<b>159</b>
D.1	Signals in pyPhysio . . . . .	159
D.1.1	EvenlySignal . . . . .	160
D.1.2	UnevenlySignal . . . . .	163
D.1.3	Segmentation of signals . . . . .	165
D.2	Algorithms in pyPhysio . . . . .	166
D.2.1	Filters . . . . .	167
D.2.2	Estimators . . . . .	168
D.2.3	Indicators . . . . .	169
D.2.4	Tools . . . . .	170
D.3	Pipelines in pyPhysio . . . . .	171
D.3.1	ECG processing pipeline . . . . .	173
D.3.2	EDA processing pipeline . . . . .	178



# List of Tables

3.1	Main physiological indicators provided by default in pyPhysio	40
3.2	Algorithms provided in pyPhysio, grouped by type of signal	40
4.1	Sessions of the experiment to create the WCS dataset. . . . .	53
5.1	Summary of main state-of-art accuracies for the recognition of arousal and valence based on the DEAP dataset. BVP: Blood volume pulse; EDA: Electrodermal activity; RESP: Respiration; ST: Skin temperature; EMG: Electromyogram, EMGz: EMG of the Zygomaticus muscle, EMGt: EMG of the Trapezius muscle; EEG: Electroencephalogram; HMM: Hidden Markov Model . . . . .	61
5.2	Source datasets, physiological signals and processing pipeline applied: ABOY [1], DBD (see Appendix A), BEN [13], PKB (see Appendix B), APK: adaptive peak detection . . . . .	65
5.3	Classification performances achieved on the DEAP dataset with LOO CV scheme according to different pipelines. . . . .	70
5.4	Classification performances achieved on the DEAP dataset with 10x5 CV scheme according to different pipelines. . . . .	71
5.5	Classification performances of four different pipelines for the WCS-Medical subset 10x5 CV scheme. . . . .	72

5.6	Classification performances achieved on the WCS-E4 and WCS-HB subsets with 10x5 CV scheme according to different pipelines. . . . .	73
6.1	Confusion matrix from internal Out-Of-Bag of RandomForest classification of infant behavioural states. . . . .	81
6.2	Summary of input data for the detection of fatigued states from inertial signals. . . . .	85
6.3	Number of samples for each subject and class. . . . .	85
6.4	Summary of input data for the detection of fatigued states from inertial signals. . . . .	87
B.1	Symbols used to indicate the main signals or signal components computed in the pipeline. . . . .	142
B.2	Values of estimated Bateman parameters for the subjects in the DEAP dataset, according to different pipelines . . . . .	151
B.3	Values of estimated Bateman parameters for the subjects in the WCS-Medical dataset, according to the three pipelines	152

# List of Figures

2.1	Representation of a Body Sensor Network within a multi-modal experiment setting with three wearable devices and two subjects. In general, a wearable device embeds multiple sensors to acquire diverse physiological signals. The smartphone, which also embeds additional sensors, serves as Body Central Unit thanks to functionalities provided by a specific app (PhysioREC logo is shown here). . . . .	7
2.2	Effects of sampling frequency on the estimated IBI series (top row), a frequency domain HRV indicator (normalized HF, middle row) and a time domain HRV indicator (RMSSD, bottom row). On the left: absolute values from the original signal (blue) and from its down-sampled version (128 Hz, red); on the right: relative differences between the two signals (note the different scale on the top plot) . . . . .	14
2.3	Overview of Physiolitix architecture . . . . .	21
2.4	Overview of PhysioREC User Interface. From left to right: login screen with authentication, activation of input and output modules, acquisition screen , real-time plots of signals.	23
2.5	Integrayion of WDs into PhysioREC . . . . .	26
2.6	Web-interface showing a list of sessions for a single experiment (Drowsiness Detection) . . . . .	30

3.1	Three steps of physiological signal processing (top) and an example on a Blood Volume Pulse signal (bottom): original signal (left), after preprocessing (middle left), result of beat detection (middle right) and computed physiological indicators of Heart Rate Variability (right). . . . .	34
3.2	Representation of a generic algorithm characterized by the computational function $F$ and its parameters $\mathbf{p}$ . . . . .	38
4.1	Illustration showing the devices used for the acquisition of physiological signals. A: Thought Technology FlexComp (picture from <a href="http://www.thoughttechnology.com">www.thoughttechnology.com</a> ) with four wired sensors: a) Electrodermal activity, b) Blood Volume Pulse, c) Electrocardiogram, d) Respiration; B: Empatica E4 (picture from <a href="http://www.empatica.com">www.empatica.com</a> ); C: ComfTech HeartBand. . . . .	51
4.2	A 120 seconds length segment of WCS signals. Blue: signals collected with medical-grade device (FlexComp); Red: signals collected with Empatica E4; Yellow: signals collected with ComfTech HeartBand. Note effects of body movements on the BVP signal of Empatica E4 due to a sudden chest movement (probably a cough) observable in the central part of the plot (respiration signal) and on the module of Acceleration from Empatica E4. . . . .	57

5.1	The ML pipeline for unbiased classification proposed within the SEQC/MAQC-III consortium [110]. In (a) the input dataset is split into 5 subsets (folds), maintaining the class stratification. A fold is kept for test, the remaining are used to train the model in the following steps. In b) a first predictor is used to optimize the parameters of the model by grid-search. An internal CV (e.g. 10x2-CV) is used to avoid overfitting. The tuned classifier is used to estimate the feature ranking (c). A second predictive model is defined (d), taking in input a reduced set of features (selected by previous ranking). Its performance is evaluated on the test set (e). The step is iterated considering an increasing number of features. The whole procedure is performed over all the folds and repeated 10 times to evaluate the overall performance of the model (e.g. MCC, Accuracy), the overall feature ranking and its stability over the iterations. . . . .	68
5.2	Overall accuracy on all datasets. 95% Confidence intervals computed by Student's bootstrapping. . . . .	70
6.1	An HRV analysis pipeline available through the Physiolyze web interface. Three processing steps (preprocessing, HRV index extraction, predictive modeling) are linked together to form the workflow in the central panel. Panel on the right allows setting module parameters: parameters for HRV index extraction are shown here. . . . .	80
6.2	Setup of the Body Sensor Network composed of 3 Inertial Measurement Units: on the back, right thigh and right leg. Left: orientation of sensors; right: positioning during a pilot test. . . . .	83

6.3	Differences in IBI <sub>mean</sub> between types of parents. . . . .	92
6.4	A frame of the video of the music-therapy session, showing two children with ASD (left and right) and the therapist (middle) while playing drums. Graphs on the left-bottom of the figure show the average IBI (RR mean) for each subject: therapist (blue), child on the right (green) and child on the left (red) during the entire session. The instant corresponding to the showed frame is identified by the vertical red line. Vertical bars on the bottom-right show current levels of synchrony: between Therapist and Child1 (left), between Therapist and Child2 (middle) and between the two children (right). Here an high level of physiological synchrony is found between the two children who are coordinated in playing drums together. . . . .	95
A.1	Overview of the DBD algorithm with the four processing stages. . . . .	126
A.2	Steps of stage 3 to identify $t_{0_i}$ . From top to bottom: a) identification of the maximum $t_{p_i}$ and $t_{p_i} - 0.25$ on the filtered signal $x_f(t)$ ; b) identification of the maximum of the derivative of the signal $x_n - x_{n-1}$ in the interval $t_{p_i} - 0.25, t_{p_i}$ ; c) identification of the local minimum of $ ds(t)/dt $ in the interval $[t_{m_i}, t_{p_i}]$ , corresponding to $t_{0_i}$ , the percussion peak in the original signal (d). . . . .	128
A.3	Performances of IBI detection on the Fantasia dataset. . .	133
A.4	BA ratio of HRV indicators on the Fantasia dataset. . . . .	134
A.5	Performances of IBI detection on the BVP from the Flex-Comp (left) and E4 (right) during baseline (top) and move (bottom) portions. . . . .	135

A.6	BA ratio of HRV indicators on the BVP from the FlexComp (left) and E4 (right) during baseline (top) and move (bottom) portions. Vertical scale is limited to the interval [0,2] for clarity. . . . .	135
B.1	Flow charts of the two novel algorithms introduced in this paper: A) peak based loss function and B) derivative-based phasic estimation. . . . .	141
B.2	Effects of wrong estimates of $\tau_1$ and $\tau_2$ on the recovery part of the resulting deconvolved signal. Top row: effects of smaller (left) and greater (right) values of $\tau_1$ ; bottom row: effects of smaller (left) and greater (right) values of $\tau_2$ . Signals have been normalized to have a maximum amplitude of 1. . . . .	144
B.3	Detrending in case of non overlapping (A) and overlapping (B) peaks. . . . .	146
B.4	Performances of the three pipeline on the DEAP dataset. By subject (left) and overall distribution (right). . . . .	150
B.5	Performances of the three pipelines on the WCS dataset. By subject (left) and overall distribution (right). . . . .	150
C.1	Wearable devices tested in this work. . . . .	156
D.1	Plot of the <code>EvenlySignal</code> created in Listing D.2 . . . . .	161
D.2	Plot resulting from Listing D.3. Note the different starting time of the two signals. . . . .	163
D.3	Plot resulting from Listing D.5 . . . . .	166
D.4	Plot resulting from Listing D.8 . . . . .	168
D.5	Plot resulting from Listing D.9 . . . . .	169
D.6	Plot resulting from Listing D.11 . . . . .	171
D.7	Plot resulting from Listing D.15 . . . . .	175
D.8	Plot resulting from Listing D.16 . . . . .	176

D.9 Plot resulting from Listing D.17 . . . . .	177
D.10 Plot resulting from Listing D.21 . . . . .	179
D.11 Plot resulting from Listing D.22 . . . . .	180

## List of abbreviations

<b>ACC</b>	Acceleration
<b>ANS</b>	Autonomic Nervous System
<b>ASD</b>	Autism Spectrum Disorders
<b>BCU</b>	Body Central Unit
<b>BLE</b>	Bluetooth Low Energy
<b>BSN</b>	Body Sensor Network
<b>BVP</b>	Blood Volume Pulse
<b>CV</b>	Cross Validation
<b>DSP</b>	Data Streaming Protocol
<b>DBD</b>	Derivative Based Detection
<b>EDA</b>	Electrodermal Activity
<b>ECG</b>	Electrocardiogram
<b>EEG</b>	Electroencephalogram
<b>GPS</b>	Global Positioning System
<b>HRV</b>	Heart Rate Variability
<b>HF</b>	High Frequencies
<b>IBI</b>	Inter Beat Interval
<b>IMU</b>	Inertial Measurement Unit
<b>LOO</b>	Leave One Out
<b>LF</b>	Low Frequencies
<b>MCC</b>	Matthew Correlation Coefficient
<b>MEMS</b>	Micro-Electromechanical Systems
<b>ML</b>	Machine Learning
<b>PPG</b>	Photoplethysmography
<b>PSR</b>	Phasic Skin Response

<b>RCO</b>	Reverse Combinatorial Optimization
<b>RMSSD</b>	Root Mean Square of Successive Differences
<b>SDK</b>	Software Development Kit
<b>SNR</b>	Signal-Noise Ratio
<b>SVM</b>	Support Vector Machine
<b>TSR</b>	Tonic Skin Response
<b>WD</b>	Wearable Device

# Chapter 1

## Introduction

Physiological signals allow clinicians to diagnose or to monitor the health status of patients, but they also provide an effective insight into the psychophysiological regulatory mechanisms that allow an individual to adapt to environmental changes and to react to external stimuli [68, 70]. Born as a tool-set belonging to medical sciences to screen the vital signs, thanks to the development of wearable technologies [85, 25], the acquisition and quantification of physiological signals is nowadays affordable in a much wider range of contexts, from personal uses to industrial and scientific research.

Indeed, wearable technologies represent a great opportunity for the scientific community to deepen the study of human physiology and behavioral responses by paving the way for a new generation of experiments in which long-term monitoring and real-life ecological acquisitions are a key aspect. In particular, a decrease in obtrusiveness on subject's behavior and daily activities is expected by adopting wearable technologies instead of clinical devices [41].

A trade-off between the two options is however undeniable. Wearable solutions have a lower cost and higher portability than medical-grade devices, which enabled their diffusion for commercial applications [118]. In

contrast, the data quality that can be achieved is lower, subject to artifacts due to body movements and data losses [129].

Several attempts have been performed to adopt wearable sensing devices for research, with substantial differences in the type of technical solution. For instance, a number of studies is focused on creating and validating novel ad-hoc device [8, 53, 27], others adopt medical-grade devices [40] or sensor platforms [26, 99]. Although these solutions can be considered wearable, as the sensing unit is a miniaturized, battery-supplied board, they still require an experimenter for setup (e.g. placement of electrodes, start of acquisition). Further, only recent devices allow real-time streaming of collected data. In addition, they require the presence of a supporting system such as: a workstation with software available for download, decode and export the data to files, software to set sensor parameters (e.g. sampling frequency, full scale range), hardware components (e.g. electrodes, sensors), communication module and protocol (e.g. Bluetooth). Therefore, they are portable but limited to specific in-lab experiments and they are not suitable for daily life applications. Instead, in this work we focus on off-the-shelf devices designed for autonomous, continuous and real-life monitoring of physiological signals which could be setup and activated by the subject itself.

In contrast to growing market availability, broad adoption of these types of wearable devices (WDs) in research [97, 108] is still limited as prevented by two main facts:

1. Existence of different solutions: many manufacturers provide diverse products which differs in terms of both technological specifications and protocols to control the device and access the data. Lack of a common standard and fragmentation of supporting tools represent an issue to researchers who look for a unified framework where to integrate multiple devices;

2. Unvalidated reproducibility: quality of acquired data is usually verified by the manufactures but seldom validation data is provided or signals compared to medical-grade device. Thus, it often remains unverified whether WD are able to reproduce scientific studies based on medical-grade devices;

The main aim of this work is to enable usage of WDs in research by addressing such key limitations. Two objectives are pursued: first is to develop a unified framework to support the acquisition and processing of signals from Body Sensor networks (BSNs), second is to assess the reproducibility of studies based on WDs, focusing in particular on an affective computing task.

To deal with existence of different devices and protocols, we developed a real-life sensing platform, composed of a software architecture to enable and control Body Sensor Networks (Physiolitix) and a Python package for signal processing (pyPhysio). By the use of these tools we were able to collect the Wearable and Clinical Signals (WCS) dataset which can be used to compare and to assess reproducibility of algorithms on signals from WDs. The experiment reproduced the DEAP dataset [67] designed for emotion recognition based on physiological signals. Based on the WCS dataset and the processing pipelines appositely developed we were able to assess, in a condition of limited movements, that WDs allow the reproducibility of state-of-art results [67, 122].

Many scientific fields can benefit from the output of this work: for instance, psycho-physiology and affective computing researchers can expand the fields of application to real-life contexts and with increasing number of participants. Clinicians and medical researcher also can rely on WDs to release home health care solutions with continuous monitoring protocols and remote assistance. We applied this framework to heterogeneous studies and scientific projects. In this work we mention in particular the detec-

tion of fatigue in skiers by inertial data, analysis of physiological response of parents to infant crying and physiological synchrony between therapist and patient affected by Autism Spectrum Disorders (ASD) during a music-therapy session.

In the following chapters we describe the technological and scientific research we conducted to develop a unified framework for acquisition, analysis and validation of signals from WDs. In particular, Chapter 2 is devoted to give the technological and scientific background about real-life sensing and describe the Physiolitix platform. In Chapter 3 we introduce the Python package *pyPhysio*, developed for processing heterogeneous types of physiological signals with both general purpose (e.g. filters) as well as signal-specific algorithms (e.g. estimators for beat detection). The experimental settings and design to create the WCS dataset are presented in Chapter 4. Chapter 5 presents the methodological approach adopted to validate WDs as able to reproduce a standard affective computing task with results. In Chapter 6 we describe four case studies in which the proposed framework has been adopted and in Chapter 7 we summarize and conclude.

## Chapter 2

# Introducing Physiolitix: a platform for real-life acquisition of physiological streams from wearable devices

In a general sense, Wearable Devices (WDs) are portable, non-invasive devices that allow the acquisition of physiological signals during daily life, with no need of external equipment [11, 85]. It is interesting to note that the application of WDs is boosted by their improved portability (increasing miniaturization and battery life) [27], as well as on new materials and layouts that improve the portability (e.g.: epidermal tattoos [8]) and decrease the invasiveness (e.g. wireless ring pulse oximeter [55]). Such new design solutions support the concept of Body Sensor Network (BSN) and the more general concept of “Body Computing” [2, 52]. The current implementation of a BSN is that of a set of sensors, which send data to a central device that manages data acquisition, synchronization and storing [130, 83, 41] on a local file-system or on a cloud service.

Beside several advantages, WDs present also technical limitations which need to be carefully considered before designing scientific experiments

based on WDs. In this chapter we thoroughly describe and discuss main technological aspects associated to WDs (see Section 2.1). We first present a list of physiological signals of interest and associated sensing technology currently embedded in commercial devices; then we analyze technical limitations and consequent issues related to signal collection. Finally, in Section 2.2 we present *Physiolitix*: the proposed solution to enable real-life sensing from multiple WDs.

In this thesis, we adopt the following standard terms:

- **Wearable device (WD)**: the technological support embedding one sensor or more and meant to be worn (e.g. t-shirt, band, smartwatch);
- **Sensor**: the technological component that measures a physical quantity and is embedded in a WD. When referring to an implementation into a wearable device, this is indicated as **Wearable Sensor (WS)**, to distinguish from the general technological component (**sensor**);
- **Multi-sensor WD**: WD that embeds multiple sensors (e.g. a smartwatch);
- **Sensing architecture**: the chain of components involved in the data acquisition process, composed of: sensor, embedding support, transmission/storage, control unit and cloud.
- **Sensor network**: a set of wearable devices used in the same data acquisition. We will focus on the concept of **Body Sensor Network (BSN)**, to develop the main notions.

## 2.1 Wearable Devices for research

In this section we illustrate the key concepts needed to understand and discuss usage of WDs for research. In particular, we start from the physio-

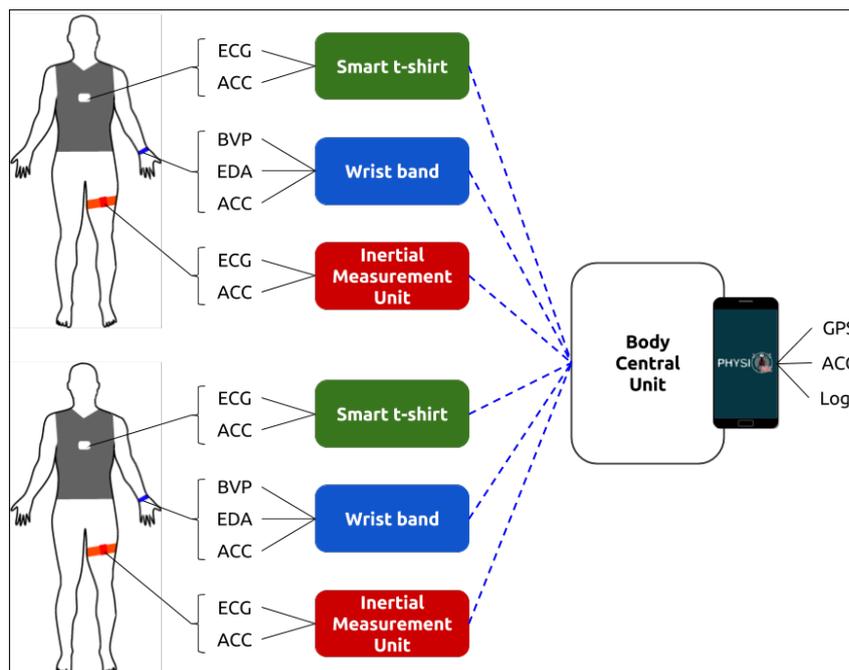


Figure 2.1: Representation of a Body Sensor Network within a multi-modal experiment setting with three wearable devices and two subjects. In general, a wearable device embeds multiple sensors to acquire diverse physiological signals. The smartphone, which also embeds additional sensors, serves as Body Central Unit thanks to functionalities provided by a specific app (PhysioREC logo is shown here).

logical phenomena that are measured (Subsection 2.1.1) to introduce more technical aspects associated to the sensing part (Subsection 2.1.2) and the issues that are addressed in this work 2.1.3.

When multiple WDs are used simultaneously, they form a BSN (see Figure 2.1). The standard BSN structure is a star network topology where the star tips nodes are represented by the WDs, the central node, called the Body Central Unit (BCU), is the device that works as a data gateway. The edges between the tips and the hub are the connections which are usually based on Bluetooth technology.

The BCU controls the nodes and receives sensor stream through the connection, ensuring that data is consistently persisted and all WDs are working correctly. This coordination capacity is provided by the app present on the BCU. In general, each WD embeds more than one sensor, thus

being able to collect more than one signal. Eventually this configuration can be replicated on multiple subjects, thus creating the more complex and more general case of BSN with multi-modal (i.e. multiple signals) and multi-subject configuration (see Figure 2.1).

### 2.1.1 Physiological signals and sensors

In this subsection we introduce the main physiological phenomena of interest for this work. For each physiological phenomenon we give a brief description of measured signal and main sensing solutions. For each signal we also provide a list of recent scientific works to illustrate the main fields of application.

#### Cardiac Activity

The cardiac muscular cells are controlled both by the branches of the Autonomic Nervous System (ANS) the parasympathetic and by the sympathetic systems. The first acts to decrease the Heart Rate (HR) and is the principal actor in resting and quiet conditions. In contrast, the complementary sympathetic system increases the HR and acts to prepare the body to face threatening situations. Heart Rate Variability (HRV) analysis is the study of the variations of the Heart Rate (HR) [76]: due to the conjunct influence of the parasympathetic and sympathetic systems, the analysis of HRV indicators is then used to investigate human emotions from a physiological perspective [68]. Two sensing technologies are currently available to acquire a cardiac signal with WDs: Electrocardiography-based (ECG-based) sensors and Photoplethysmography-based (PPG-based) sensors. ECG-based sensors detect the electrical potential generated during the contraction of the cardiac muscular cells. The signal acquired with a ECG-based sensor is called Electrocardiogram (ECG). This is a mature technology, but it lacks convenient portability, because of the need of wearing a chest-band under

the clothes. To overcome these problems, PPG-based sensors have been recently introduced as embedded in wrist-band devices or smart watches. These devices are equipped with pairs of Light Emitting Diodes (LEDs) and photo-diodes to detect the variation of blood volume in vessels through variations in absorbed light. As the blood volume varies due to heart contraction, it is possible to observe the heart activity. Wearing these devices is much more comfortable than chest bands, but the acquired signal can be more sensitive to moving artifacts [129]. The signal acquired with the PPG-based sensor is called Blood Volume Pulse (BVP).

Several studies used HRV analysis to investigate autonomic response, for instance to detect stress [30] and cognitive workload in different conditions [78, 123]. Esposito et al. [40] studied the calming effects induced in infants by being carried or held by the mother. Parasympathetic-related HRV indicators were significantly higher during carrying than during holding, suggesting that infants are more relaxed in the first condition. Joosen et al. [60] analyzed how the HRV of adults is influenced by the crying of infants showing that effectiveness of this influence is related to maternal sensitivity.

### **Electrodermal Activity**

The Electrodermal Activity (EDA) is the variation of skin conductance. It depends on the activity of the sweat glands which are under the control of the Sympathetic Nervous System (SNS). During stressful situations the SNS elicits the sweating activity which in turn increases the skin conductance that originates the measured signal [14, 104]. EDA comprises two components: the Tonic Skin Response (TSR) and the Phasic Skin Response (PSR). The TSR component reflects the physiological changes in the secretory activity of the sweat glands. The PSR component is associated to the activity of the SNS. When a subject is emotionally aroused, the

waveform presents a peak due to the sweat gland activation, which overlaps with the TSR component. The main signal processing approach to investigate EDA in association to ANS activation is possible by decomposing the TSR and PSR components [13, 6, 4]. EDA is measured by applying an electric potential between two electrodes which are in contact with the skin. The measurement is highly sensitive to body movements, because a subtle drift in the placement of sensors could expose a new area of the skin to the contact with electrodes. The new skin can be considerably dryer as it has not been previously in contact with electrodes. This could affect the measurement of the EDA introducing sudden changes in the baseline level. EDA sensors embedded in wristbands can be sensitive to this type of noise. Clinical sensors use straps or other solutions to fix electrodes to the fingers to be able to reduce sensor movements.

Being related to the sympathetic branch of the ANS, EDA has been extensively used to investigate the psycho-physiological response to external stimuli. Hernandez et al. [54] used EDA analysis to investigate the engagement of children during interactive playing session with adults. Ruiz-Robledillo et al. [106] showed that caregivers of relatives with Autism Spectrum Disorders (ASD) present lower EDA levels in response to stressful events, suggesting the existence of an adaptation mechanism that allows caregivers to habituate to stress. Also, EDA-derived features have been extracted and used to detect fatigue during driving [20] and to discriminate stress from cognitive load in an office work related environment [111]. By analyzing the EDA, Sano et al. were able to predict the onset of epileptic seizures [89].

### **Brain activity**

Brain activity is measured by detecting the electrical potential generated by the ionic currents in neurons. Collected signal is called Electroencephalo-

gram (EEG) and typically multiple electrodes are placed in different points of the scalp to observe activation of specific areas of brain cortex. Each electrode is referred to a specific brain area which is associated to a particular brain function, allowing the analysis of brain activity and response under different stimuli and tasks. The EEG signal is processed to extract features able to measure the brain activity, for instance power in the brain wave bands, fractal measures and Event Related Potentials (ERPs). The complexity of the sensing equipment needed to acquire good quality signals limits the number of possible scenarios in which the EEG can be used. However, new portable EEG devices with limited number of electrodes are being introduced and allow raw measurements of EEG in daily life [102, 7].

The analysis of the brain activity has been used to identify and classify emotions. Frantzidis and colleagues [43] extracted the spectral components of EEG and amplitude and delay of detected ERPs to classify four emotional states elicited by different visual stimuli. A similar approach has been used by Jiang et al. [59], which integrated also the acquisition of eye activity and facial expressions. Liu et al. [74] used Fractal Dimension of EEG to define a subject-specific framework to classify emotions in real-time. Results of the emotion recognition from EEG data have supported the use of EEG as a potential communication channel to interact with machines and communicate emotions [45].

### **Physical activity**

Activity monitors acquire inertial (acceleration) and spatial (rotation, position) signals to measure body movements and human physical activity, and are called inertial monitoring units (IMUs). The sensing component is based on microelectromechanical systems (MEMS) sensors, which provide highly miniaturized solutions with relatively low energy requirements. Therefore, activity sensors are commonly embedded in smartphones and

many other powered devices. IMUs have been extensively adopted in scientific studies, in particular to analyze movement and the ambulatory process [23]. They are also systematically considered to track the physical activity of subjects in order to reach and maintain an active lifestyle [15]. Stereotyped motor movements [92] are investigated to provide an index of severity of Autism Spectrum Disorders.

### **Other physiological phenomena**

Those listed in the previous subsections are the main physiological phenomena commonly measured by WDs. However, other phenomena of interest can be measured, either by specific WDs or by applying appropriate algorithms to derive new measurements from the collected signals. For instance, advanced PPG sensors do also provide measurement of hemoglobin saturation [55]. Breathing activity can be measured by acoustic sensors or extracted with appropriate algorithms from the chest movements measured by an IMU [69] or from cardiac signals, such as the BVP [75]. As respiration patterns influence the cardiac activity and frequency of the heart rate, adding the information regarding the breathing activity to the study of HRV allows for a better insight into the autonomic regulation. For instance, Valenza and colleagues [124] were able to improve the performance of an emotion recognition quadratic discriminant classifier by embedding cardio-respiratory coupling into the set of input features. WDs have also been proposed to measure the muscular activity by Electromyographic (EMG) sensors, for instance to recognize emotions from facial expression [67] or hand gestures [62]. Further, recent trends in sensor technology and research aim at integrating measurements of bio-chemical components on the skin and physiological signals [57, 8].

### 2.1.2 Technical aspects

Current technological research aims at increasing both miniaturization and portability of sensors. On one side this enables easy-to-wear devices, signal acquisition in ecological conditions and decreases obtrusiveness. As a major drawback, miniaturization is a compromise in terms of signal efficiency, thus of signal quality (signal noise ratio). In this subsection we consider the main technical characteristics and constraints of WDs for physiological signals and analyze their effects on the quality of the collected data.

#### Sampling frequency

All WDs are battery supplied. To increase autonomy, performances of some sensor components are reduced to decrease battery consumption. In particular, the sampling frequency is kept lower than medical-grade devices, usually below 256 Hz. Also, performances of other components are reduced, for instance LEDs light intensity in optical sensors and gain of amplifiers.

The sampling frequency ( $f_{samp}$ ) determines the maximal frequency component that can be observed in the signal (also known as Nyquist frequency). Further it also defines the minimum temporal resolution  $\Delta t = 1/f_{samp}$  to discriminate events. For instance, for HRV analysis it is recommended [76] to measure the distances between subsequent beats (or Inter Beat Intervals, IBIs) with a resolution of milliseconds, hence requiring a minimal sampling frequency of 1000 Hz, far more than what usual wearable sensors can provide. In some cases this limitation can be overcome by over-sampling the signal at the processing stage. Figure 2.2 shows the effect of lower sampling frequency on the computed IBI signal and HRV indicators.

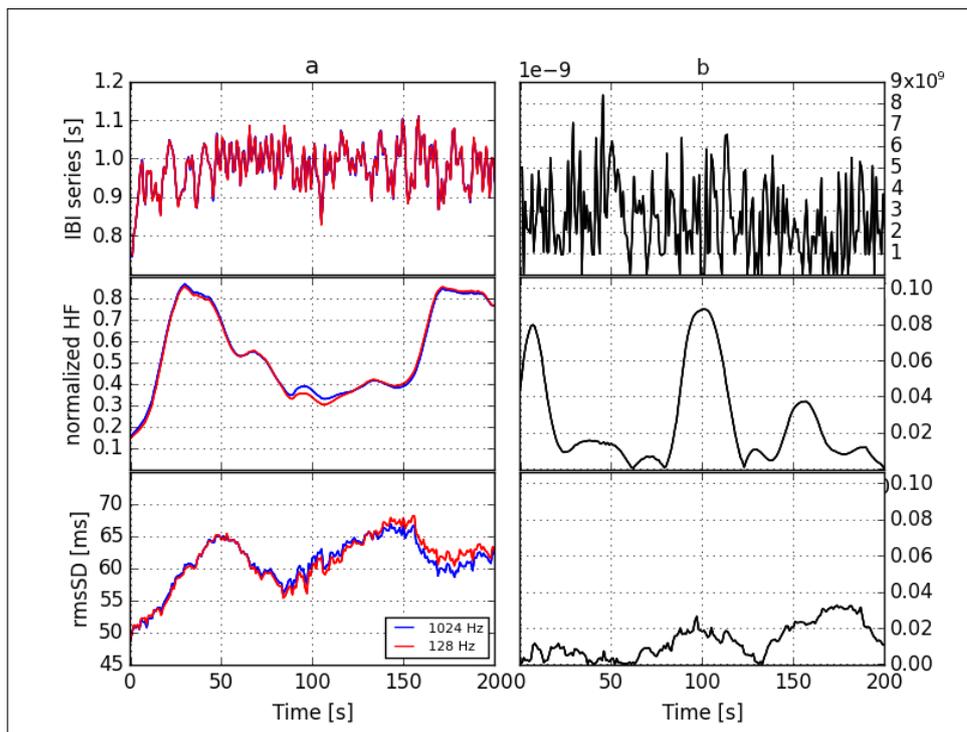


Figure 2.2: Effects of sampling frequency on the estimated IBI series (top row), a frequency domain HRV indicator (normalized HF, middle row) and a time domain HRV indicator (RMSSD, bottom row). On the left: absolute values from the original signal (blue) and from its down-sampled version (128 Hz, red); on the right: relative differences between the two signals (note the different scale on the top plot)

**Embedding and sensor positioning**

In order to favor wearability, some technical constraints are imposed, thus WDs generate signals which have lower signal-noise ratio (SNR) than their medical-grade counterpart. First, usually in WDs no aids are used to fix the sensors to the body (e.g. adhesive ECG electrodes); instead they are kept on position only by the elastic properties of the embedding support (e.g. tissue of T-shirts, rubber band of bracelets).

However, minimal shifts of the sensors due to body movements that can introduce noise on the acquired signal such as drops or sudden value changes. Signal quality can also be influenced by a non optimal positioning of the sensor: although some anatomical loci are more appropriate than others for measuring the signal, the need of embedding the sensor in a wearable and comfortable support might constrain the positioning to a non-optimal locus where the signal magnitude might be lower, such as the wrist instead of hand palm in the case of EDA [125]. Artifacts might be generated by high or low dynamic body movements. High dynamic movements (sudden movements and/or with high amplitude) may cause the displacement of the sensors that are not strictly fixed to skin surface. Artifacts appear as impulsive changes in the signal value that might cause saturation or loss of signal. In the case of low dynamic movements there is no displacement of the sensor but the physiological measure is influenced by inertial effects of body movements; in particular physiological blood fluid dynamic is particularly sensitive to limb movements and changes in body positions [87, 129, 126].

**Autonomy**

Capacity of internal memory should not affect the performances of a WD: as a first solution as the signals can be sent to the BCU and not locally

stored. However, some devices can also store data as a backup, thus would need a large capacity for prolonged and high sampling frequency acquisitions. More in general, lower energy requirements and battery capacity directly affects the autonomy of the device and, consequently the duration of the acquisition. High capacity and rechargeable batteries would represent the ideal solution, however their size is bigger and together with recharging circuits they might increase substantially the dimension of the WD. The choice of the streaming technology (e.g. by wireless connection) determines further constraints to autonomy of the sensing architecture. Bluetooth is the most common technology for WDs; only Bluetooth Low Energy (BLE) allows significant reduction of battery consumption and extended the range. However, BLE is a quite recent technology and is not extensively supported.

### **Signal quality characteristics**

Technical characteristics account only partially in the evaluation of WDs. Indeed, it is important to consider also the quality profile of signals acquired with WDs. A first quality indicator is the signal stability, which can be used as an overall estimate of the WD reliability. Several technical causes (e.g. sensor disconnection, errors in the streaming protocol) can contribute to affect the acquisition, which results in missing or incomplete data. Second, it is fundamental to consider the sensitivity of the WD to body movements and other causes that might reduce the SNR. A complete evaluation of a WD should include testing for SNR in a typical experiment. In summary, there are no general indicators for signal stability and sensitivity able to quantify and compare performances of different WDs. Instead, this evaluation should consider the specific needs of operating the WD in different conditions where it is expected to be applied.

### 2.1.3 Issues intrinsic to WD usage

Technical constraints described in the previous subsection (2.1.2) are intrinsic to physiological sensing based on WDs. Consequences of these limitations can be partially addressed only by specific signal processing techniques or adequate design and execution of the experiment. Further, in this subsection we discuss three limiting factors that pertain to the usage of WDs, independently of the type of sensor or technical characteristics:

1. Integration of different Data Streaming Protocols;
2. Signal synchronization;
3. Privacy and security.

Addressing and solving these three issues is the aim of the *Physiolitix* platform proposed in Section 2.2.

#### **Middle-/software characteristics**

WDs are integrated in BSNs, up to fairly complex architectures, needed when experiments involve multiple devices or more than one subject. Typically, each WD adopts a specific method to interface with the BCU or other devices. We call this method Data Streaming Protocol (DSP) and we intend the set of specifications that allow the BCU to communicate with the WDs and interpret received messages.

The DSP can take different forms depending on the manufacturers of the WDs, but in general it is often provided to allow researchers and developers to build personalized software to be installed on the BCU to control the WD. Alternatively, a proprietary app is usually provided with minimal functions.

Depending on the commercial strategies the DSP is made available as a Software Development Kit (SDK) or protocol documentation. A SDK is

a set of tools for (i) development of software with a documented library to manage the communication with the WD and (ii) development of a custom interface to control the data acquisition. The protocol documentation, instead, is the set of codes adopted in the streaming messages with the corresponding meanings. It is provided to allow the development of middle-layer software to correctly interpret the stream. Since every manufacturer adopts different methods of delivering the data, a specific software solution must be implemented for every protocol and SDK.

In summary, while on one side the DSP should facilitate access to sensor data and control of acquisition, on the other side, the existence of numerous solutions complicates the integration of the DSP, which is one of the major issues for the development of a working sensing architecture for research use.

### **Synchronization**

Signals acquired simultaneously by the same BSN are likely going to be analyzed together in order to contribute to the investigation of the same phenomenon. This approach is usually called Data Fusion: for instance accelerometer data are often used to improve the BVP signal quality by canceling the noise caused by body movements [47, 119]. Results from Data Fusion techniques might be dramatically affected by temporal misalignment and it is thus fundamental that all acquired signals are time-synchronized before any processing step. We call synchronization the process of assigning to each sample in a signal a temporal reference called 'timestamp'. Hence, samples from different signals with the same timestamp can be assumed as they were generated simultaneously [16].

Again, the solution of the synchronization problem is complicated by the existence of different technological implementations. Three main cases can be identified:

1. The WD embeds an internal clock which is used to provide an absolute time reference and the clock can be set manually;
2. The WD embeds an internal clock which is used to provide an absolute time reference but the manual set is not available;
3. No clock is embedded and the timestamp, if given, is represented by a counter of generated samples, thus providing only a relative time reference.

Except the third case, the synchronization would be possible by estimating the time offset between each device. The time-offset between two WDs might not be constant due to minimal variation in the clock pace; the ratio between the pace of two clocks is called time-drift [112]. However, in the case of short acquisitions this effect can be negligible and the time-drift can be considered equal to 1.

A third technical element that prevents the correct synchronization is the latency between the generation of the sample and the assignment of the timestamp. Latency is affected by delays due to the initialization of the sensor, to the communication mean or to the processing of the message (both on the WD and BCU sides). Worst-case scenario is when the WD does not embed a clock and the assignment of the timestamp takes place on the BCU, as in such case all the causes contribute to increase latency.

### **Privacy and security**

Physiological signals are sensitive personal data that require a specific treatment in terms of data management and control of access. The use of commercial applications provided by WD manufacturers represents a serious threat to personal privacy as they allow no control on who has access to the data. In particular, it is quite common that the acquired signals are automatically sent to a third-party proprietary server where

they are stored and no control is left to the researcher who often is only able to download the data. At the same time there is no control on the procedure of data encryption. This aspect poses a significant obstacle to the use of WDs for research as the experimenter is not able to guarantee the protection of personal data and security.

#### 2.1.4 Existing platforms

Notwithstanding the importance of WDs for research, a clear solution to integrate physiological devices into a unique platform is still missing. Currently, the research is focusing on the general problem of integrating multiple sources of data connected to the internet, which compose of what is known as the Internet-of-Things (IoT). One of the main goals is to seamlessly connect the many resources to a cloud service, in order to allow real-time data analytics to generate new valuable information. Hence, is then used as a feedback or a decision trigger [96]; however, the lack of a standard protocol for communication and the existence of diverse technological solutions is still a major issue [35, 79, 18].

The solutions currently available are mainly represented by commercial products. Although different in terms of features and scopes, they all share the same key strategy of providing API<sup>1</sup> for each specific device while exposing a common interface to the developer.

In the medical and health-care field, current state-of-art is still focusing on the identification of optimal architectures or paradigms regarding implementation, but a working solution is still missing [103, 50]. In [28] Chen and colleagues define various guidelines for the next generation of WDs, which is projected to be based mainly on smart garments. In particular, they propose to structure the architecture into three layers: (a) the front-end or User Interface, to monitor the acquisition and provide feedback, (b)

---

<sup>1</sup>see, for instance: [reference.humanapi.co](http://reference.humanapi.co)\newline<http://getthehealth.io>\newline[validic.com](http://validic.com)

the communication mean, centred on a smartphone and (c) the backend represented by the cloud solution.

These three layers are exactly the three blocks that compose Physiolitix, our solution for the integration of multiple WDs for research. Physiolitix set out to represent one of the first working open-source platforms that aim at solving the issues intrinsic to the WDs usage presented in the previous section.

## 2.2 Result: Physiolitix

Physiolitix is a cloud-based framework to acquire, store and manage physiological signals from WDs. Its main characteristic is flexibility in terms of input sensors and application.

The Physiolitix architecture is composed of two main sections (see Figure 2.3):

1. **PhysioREC**: Android app for real life data acquisition;
2. **PhysioBackend**: cloud resource for the storage, processing and web-access of experimental metadata and sensor data.

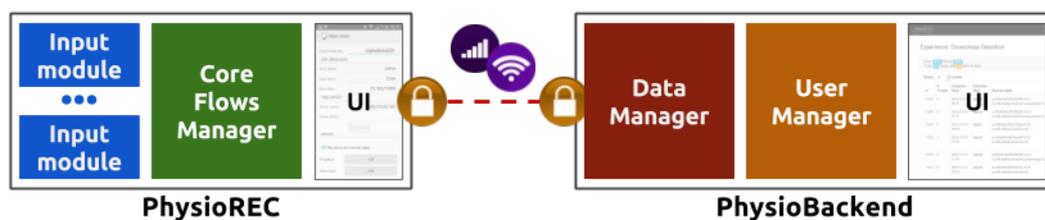


Figure 2.3: Overview of Physiolitix architecture

The PhysioREC data acquisition section is organized as BCU to which Bluetooth-enabled devices can send data. The generic BCU device can be a smartphone or a tablet. PhysioREC is the Android application that

manages the sensor data collection, synchronization and stream. It also takes care of encrypting, compressing and sending data through a Wi-Fi or 3G/4G data connection to PhysioBackend. There, customizable pipelines can be applied in real-time to data for downstream analysis, in particular for extraction of physiological metrics and indicators. In the following subsections we describe in detail the two components.

### 2.2.1 PhysioREC: real-life sensing

The PhysioREC Android app is the first element of Physiolitix architecture. Its main goal is to provide a unified platform for sensor data acquisition in multi-device experimental settings where two main issues arise: (a) the existence of different data streaming protocols and (b) the synchronization of signals from different sources. It also addresses the collateral problem of (c) privacy of personal data and security of the data transmission.

Key components of PhysioREC are:

1. Input and output modules;
2. User interface;
3. A Core Flows-Management function.

#### Implemented Input and Output module

The Input and Output modules operate as interfaces to the variety of existing DSPs and WDs. An Input module is specifically developed for each WD based on the existing method (SDK, protocol documentation) provided by the manufacturer to allow the connection between smartphone and WD and the control of the data acquisition. The Input module translates the proprietary method into a set of predefined functions that are then exposed to the Core Flows Management. Similarly, Output modules

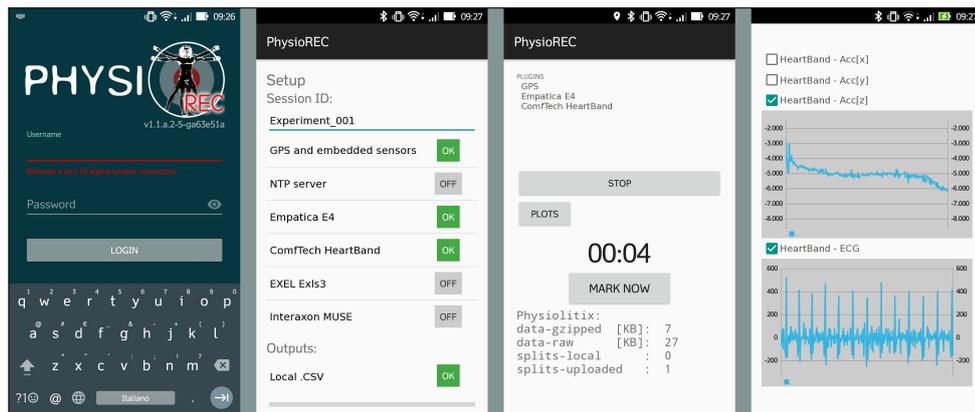


Figure 2.4: Overview of PhysioREC User Interface. From left to right: login screen with authentication, activation of input and output modules, acquisition screen , real-time plots of signals.

allow the stream of the acquired data to different destinations, such as a database or a real-time processing cloud service.

The Input modules developed so far include:

- **AndroidDevice**: to record logs, performance data and inertial signals from sensors embedded on the smartphone;
- **EmpalinkDevice**: allows the management of the Empatica E4 (connection, interaction, setup and data collection). It is based on the Empatica SDK Empalink v2.0;
- **ComfTechDevice**: allows the management of the ComfTech smart-garments series, such as the HeartBand or the CozyBaby. It is based on the documentation of the hexadecimal communication code provided by the manufacturer (ComfTech);
- **ExlsDevice**: allows the management of the IMU EXEL Exls3 board, based on the documentation of the hexadecimal code provided by the producer (EXEL);
- **InteraxonMuseDevice**: allows the management of the wearable EEG Interaxon Muse. Based on the SDK provided by Interaxon;

- NTPserver: to connect to an external server providing a time reference signal through Network Time Protocol.

Available Output modules currently implemented are:

- PhysiolitixOutput: to connect and stream data to the Physiolitix backend which is the main output of PhysioREC. This module implements methods for data chunking, compression and local caching. The policies to regulate the data preprocessing and streaming are automatically defined to optimize the battery consumption and utilization of data connection.
- CSVOutput: allows saving the acquired data locally in human readable comma separated values (.csv) files.

### **User Interface**

The PhysioREC User Interface (UI) is composed of three parts (see Figure 2.4):

1. Login screen to allow the user authentication and the selection of the experiment associates to the acquired data;
2. Experimental setup: shows the list of Input and Output modules that can be activated and allows the insertion of experimental metadata;
3. Acquisition progress: shows information during the acquisition (such as: connection status, amount of data, errors). It also allows the annotation of the sessions through a marker and the inspection of the quality of the signals.

### **Core Flows Management**

The Core Flows Management is the basic component of PhysioREC and it is responsible for the coordination of the Input and Output modules, the

signal preprocessing and the management of the acquisition. It is built on top of SensorFlow [12], a Java Library for the advanced management of data streams.

Through the Input modules, the Core Flows Management activates the connection with the WDs and manages the start and stop of the acquisition session. Collected data are synchronized by assignment of a unique time reference, provided by an external source or by the smartphone clock; then the preprocessed data are sent to the Output modules. A key feature of the Core Flows Management is the persistence of the collected data: a local database is used to temporarily store the data which are deleted only upon confirmation from all the Output modules that transmission to their destinations is complete. In this way, PhysioREC is robust to loss of data connection or errors due to sensor disconnections.

### **Advantages of PhysioREC**

PhysioREC provides a solution to (a)(b)(c) issues identified in multi-device experiments with WDs. The existence of specific data streaming protocols for each WD prevents the adoption of settings where different WDs are used simultaneously, as this would require to run the proprietary app of each device and separately control the acquisition. On the other hand, it would not be feasible to develop a specific app for each experiment setup. Due to the speed at which new WDs are released or the protocols updated, this scenario would require a continuous effort to keep the pace with the technological progress.

Instead, the modularity of PhysioREC in terms of inputs and outputs allows high flexibility in supporting different types of WDs while maintaining reliability and robustness of the basic functions of the Core Flows Management. In fact, the workflow to integrate a new device is essentially focused on the development of the appropriate Input module. This step

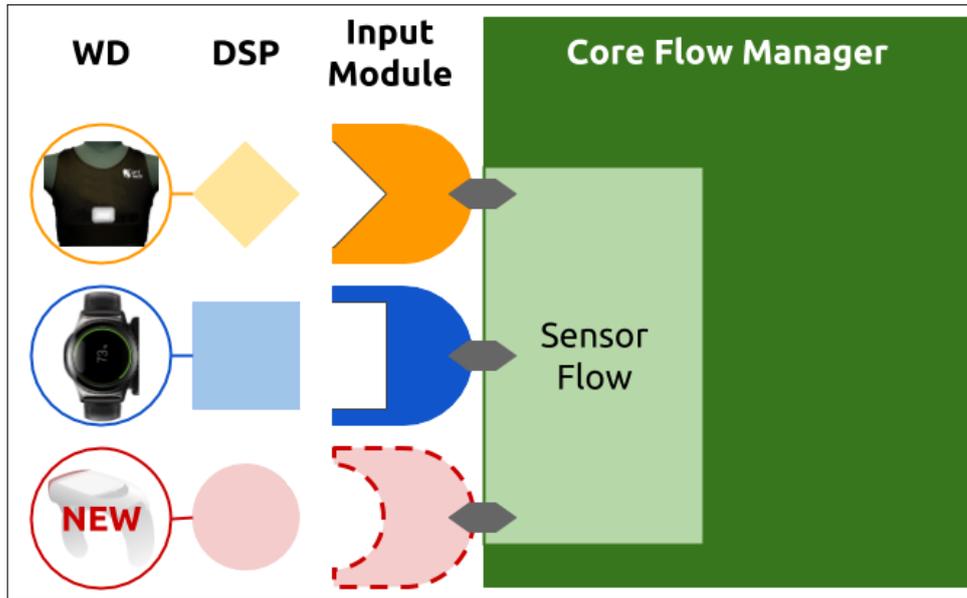


Figure 2.5: Integration of WDs into PhysioREC. From left to right: each WD which provides a Data Streaming Protocol (DSP) can be integrated by developing the appropriate Input module. The Input module works as an interface between the specific DSP and SensorFlow by embedding the DSP into a set of defined functions, which can be used by the Core Flow Manager to communicate with the WD through SensorFlow. Therefore, when a new device is available, only its Input module needs to be developed to fully exploit the other features and characteristics of the Core Flows Management.

(see Figure 2.5) aims at developing a middle-ware layer that translates the functions provided by the DSP of the new device into the general functions of SensorFlow, to allow the Core Flows Management to access the device and collect its data.

In addition, the use of PhysioREC is not limited to the physiological WDs considered in this analysis: it would indeed be possible to use PhysioREC to acquire data from other sources in the Internet of Things (IoT), for instance from environmental sensors. Similarly, with appropriate Output modules the data can also be used to stream the data to a third-party service, for instance for health monitoring.

Also, issues associated to the synchronization of the data flows are solved, thanks to the structure of PhysioREC. In fact, the Core Flows

Management works as unique data gathering hub and therefore it is possible to assign a unique time reference to each sample, corresponding to when the sample is received. However, this solution is still ineffective in dealing with errors due to latency and time drift. These problems cannot be solved runtime and the optimal synchronization is performed during signal processing.

PhysioREC affronts key concerns associated to the privacy of personal data and security. In particular, we focused on the following issues to grant the security of data streaming:

- a Control of the data storage: data are not sent to third-party servers (as it usually happens when using proprietary apps) but only to the selected destination server or they are stored locally;
- b Security of the data streaming: the `https` protocol<sup>2</sup> is used to send the data to the Physiolitix backend. In addition, on-going development aims at enabling the encryption of the data before the streaming;
- c Control of the access: authentication to the Physiolitix Backend for data uploading and downloading is based on the OAuth2 standard<sup>3</sup>.

Regarding privacy, as the requirements depend on the type of subjects and experimental design, the definition of rules and procedures for each specific use-case is left to the researcher. As a general rule, to prevent the identification of the user, the personal data are never stored by PhysioREC. Additional features are currently being developed to further improve data anonymization: the automatic removal of absolute timestamps from the data and the random assignment of subject identifiers.

It is worth noting that the described restrictions and procedures have been defined to enable an environment for research purposes: in case of

---

<sup>2</sup><https://en.wikipedia.org/wiki/HTTPS>

<sup>3</sup>[https://en.wikipedia.org/wiki/OAuth#OAuth\\_2.0](https://en.wikipedia.org/wiki/OAuth#OAuth_2.0)

commercial applications additional requirements should be met and specific solutions should be implemented, for instance allowing users to control own personal data.

### 2.2.2 PhysioBackend

While PhysioREC is the Physiolitix front-end, created to be used on-the-field to collect the physiological signals, PhysioBackend is its server-side (actually cloud) platform, responsible for the management of the experiments and downstream analysis of the collected data. PhysioBackend has been developed with the following aims: (i) having a centralized place where data can be stored and accessed; (ii) granting security and privacy by controlling the access to the data

The key elements to meet the two goals are:

1. Data indexing and web interface;
2. Authentication and user-based access control.

The implementation of PhysioBackend is currently based on standard back-end platform composed of a PostgreSQL<sup>4</sup> server as database management system and Django<sup>5</sup> for the web-interface and the control of automatic data preprocessing routines.

#### Data indexing and web-interface

The data streams from the active PhysioREC instances are first temporally stacked in a transition table, then indexed and correctly stored by a Django routine. The transition table is currently part of the PostgreSQL server, but it could better be implemented in a non relational database to improve scalability. Data are indexed in sessions and experiments: a

---

<sup>4</sup><https://www.postgresql.org/>

<sup>5</sup><https://www.djangoproject.com/>

session is the set of data acquired during a single acquisition event, while an experiment is associated to the study or aim for which the data are used and is composed of multiple sessions. Each experiment is associated to a list of users that are enabled to create new acquisition sessions.

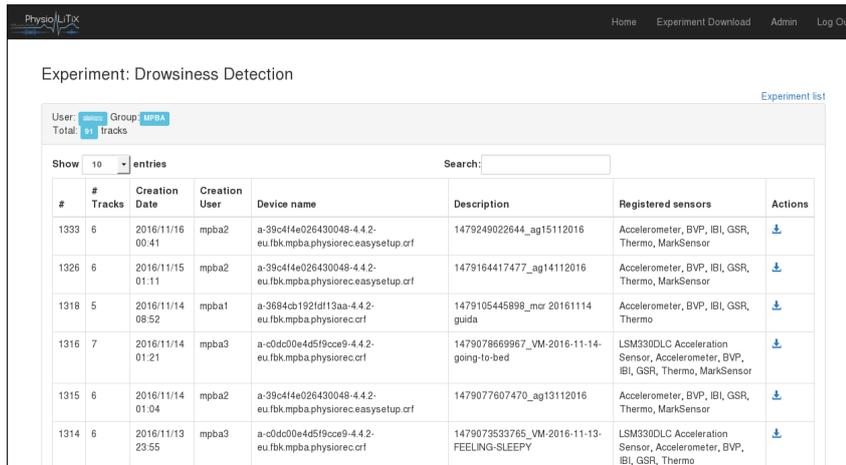
A web interface based on Django allows accessing to the data (see Figure 2.6). Upon authentication, the web-interface shows the list of owned experiments that can be downloaded or further inspected to see the list of sessions belonging to the experiment. Each session can then be downloaded separately, or further inspected to download the single signal. Metadata information, such as the name of the user, date and time of the acquisition start, type of signals acquired, is also shown and can be used to order and search the list of the results. The method adopted to store and index the data and the web-interface provide a direct and centralized access to the experimental data, overcoming the need of collecting the data from the different devices (local memory) or third-party web-services.

### **Authentication and user-based access control**

Only authenticated users can access and stream data on PhysioBackend. Authentication is provided through the PhysioREC Login interface or through the web-interface and is based on a OAuth2<sup>6</sup> authentication server. Two different types of users are conceived: the 'researcher' and the 'experimenter'; the 'researcher' user can create new experiments and 'experimenter' users, define specific owned users for the data acquisition for an experiment and access its data. The 'experimenter' can use PhysioREC to collect the data: they will be able to authenticate onto PhysioBackend to get a list of experiments for which they have been enabled and stream new data from those experiments, but they will be unable to access the collected data. With this hierarchical structure it is therefore possible to

---

<sup>6</sup><https://oauth.net/2/>



#	# Tracks	Creation Date	Creation User	Device name	Description	Registered sensors	Actions
1333	6	2016/11/16 00:41	mpba2	a-39c414e026430048-4.4.2-eu.fbk.mpba.physiorec.easyselup.crf	1479249022644_ag15112016	Accelerometer, BVP, IBI, GSR, Thermo, MarkSensor	<a href="#">↓</a>
1326	6	2016/11/15 01:11	mpba2	a-39c414e026430048-4.4.2-eu.fbk.mpba.physiorec.easyselup.crf	1479164417477_ag14112016	Accelerometer, BVP, IBI, GSR, Thermo, MarkSensor	<a href="#">↓</a>
1318	5	2016/11/14 08:52	mpba1	a-3684cb1921df113aa-4.4.2-eu.fbk.mpba.physiorec.crf	1479105445898_mcr_20161114_guida	Accelerometer, BVP, IBI, GSR, Thermo	<a href="#">↓</a>
1316	7	2016/11/14 01:21	mpba3	a-c0dc00e4d5f9cce9-4.4.2-eu.fbk.mpba.physiorec.crf	1479078669967_VM-2016-11-14-going-to-bed	LSM330DLC Acceleration Sensor, Accelerometer, BVP, IBI, GSR, Thermo, MarkSensor	<a href="#">↓</a>
1315	6	2016/11/14 01:04	mpba2	a-39c414e026430048-4.4.2-eu.fbk.mpba.physiorec.easyselup.crf	1479077607470_ag13112016	Accelerometer, BVP, IBI, GSR, Thermo, MarkSensor	<a href="#">↓</a>
1314	6	2016/11/13 23:55	mpba3	a-c0dc00e4d5f9cce9-4.4.2-eu.fbk.mpba.physiorec.crf	1479073533765_VM-2016-11-13-FEELING-SLEEPY	LSM330DLC Acceleration Sensor, Accelerometer, BVP, IBI, GSR, Thermo	<a href="#">↓</a>

Figure 2.6: Web-interface showing a list of sessions for a single experiment (Drowsiness Detection)

grant security and privacy while allowing the use of Physiolitix to multiple researchers and different studies.

## 2.3 Discussion

WDs represent an opportunity for research as they enable the collection of physiological signals in real-life context, thus facilitating the investigation of phenomena that hardly can be reproduced with laboratory settings. However, before an extensive use for scientific purposes can be achieved, some technical aspects need to be addressed and solved, in particular appropriate technological platforms need to be created.

In this chapter we introduced the Physiolitix platform. The objective of Physiolitix is to enable the adoption of reliable and robust multi-device experimental settings in real-life contexts. PhysioREC, the front-end component of Physiolitix, allows the synchronized acquisition of data from multiple WDs. Input and Output modules mask the heterogeneity of data streaming protocols and the Core Flows Management provides features such as synchronization, fault tolerance and robust transmission. The

PhysioBackend is built to centralize the streams and manage the acquisitions, while exposing to researchers essential functions to plan the experiments, download data and control the access to sensitive information.

Although being stable and already used in international research projects, Physiolitix can be improved. Further development is needed to increase the number of compatible WDs and more functionalities are required, such as the remote creation of experimental settings and diagnostic tools to inspect the signal quality. To this aim the source code of Physiolitix is made available to contributors upon request.

Assessing the technical feasibility and enabling the real-life sensing with multiple signals is however not sufficient to grant adoption of WDs in research. Signals acquired by WDs have specific characteristics associated to technical limitations and constraints which need to be addressed with appropriate signal processing techniques. In addition, WDs allow the simultaneous observation of different physiological phenomena: for instance, Heart Rate Variability, Electrodermal Activity and Respiration. As for the upstream part, also the downstream analysis requires a unified framework for the extraction of information of interest from the collected signals. The solution proposed in this work is the Python package `pyPhysio`, which is introduced in Chapter 3.



## Chapter 3

# pyPhysio: Physiological signal processing with Python

The importance of physiological signals to decode the psychophysiological state [68], combined with the emergence of wearable technologies which ease the acquisition of physiological signals in real-life, are the main reasons to require for new software resources to process heterogeneous physiological signals. In general, available resources, both commercial and open-source, focus only on a single type of signal, such as pyEEG [9] for electroencephalographic (EEG) signals, pyHRV [17], gHRV [100] and KUBIOS [120] for Heart Rate Variability (HRV) analysis; Ledalab [13], SCRalyze [5] and cvxEDA [51] for electrodermal activity (EDA). A collection of available functions for physiological signal processing is also provided by the PhysioNet [49] website. Such fragmentation of solutions in multiple software tools is one hurdle for reproducibility and comparison between studies, as researchers usually have to develop custom code to include in a unique study different types of signals. Such code is also of critical importance for reproducibility.

Further, due to partial availability of open-source tools for physiological signal processing, usually psycho-physiology researchers rely on commercial software, which implements black-box algorithms, thus preventing

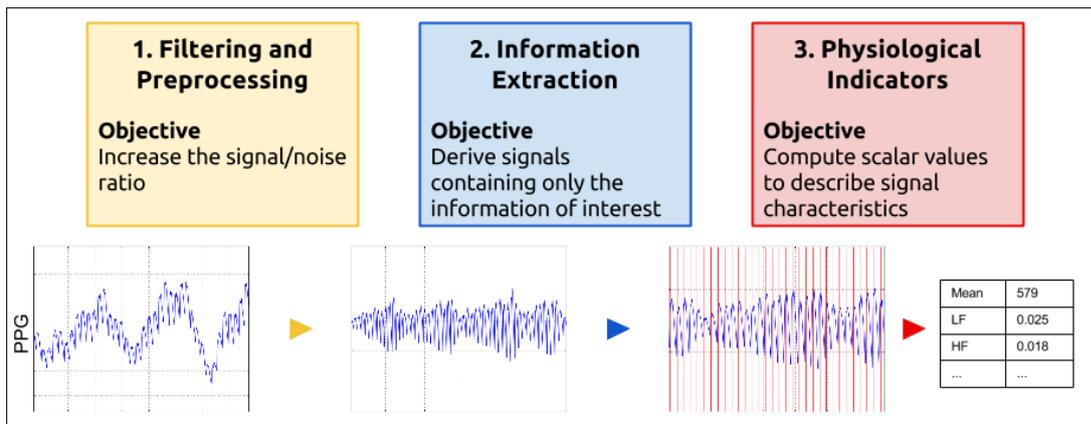


Figure 3.1: Three steps of physiological signal processing (top) and an example on a Blood Volume Pulse signal (bottom): original signal (left), after preprocessing (middle left), result of beat detection (middle right) and computed physiological indicators of Heart Rate Variability (right).

comparisons and reproducibility.

In this chapter we introduce `pyPhysio`, a python library to process physiological signals, targeting multi-modal experimental settings and psychophysiological analysis. The remaining part of this chapter is organized as follows: Section 3.1 presents few theoretical concepts about physiological signal processing, Section 3.2 describes how these concepts have been implemented in `pyPhysio` by providing the details of the library structure. In Section 3.3 we consider each class of physiological signals for which `pyPhysio` provide a specific pipeline. In Section 3.4 we summarize and conclude, suggesting possible contributions and next steps.

## 3.1 Physiological Signal Processing

In this section we briefly introduce few general concepts about signal processing that drove the development of *pyPhysio* and are useful to better understand the structure and correct use of the library.

A standard signal processing procedure can be decomposed into the following three stages (see Figure 3.1):

1. Preprocessing;
2. Extraction of information;
3. Computation of physiological indicators.

Specific applications, such as network inference and deep learning, could have a different final stage, for instance:

- 3b. Computation of similarity metrics between signals or convolution.

**Preprocessing** aims at increasing the Signal Noise Ratio (SNR): the information content after preprocessing is the same as before, but undesired components are attenuated. For instance, this stage includes operations such as filtering out noise in high frequency bands, removing trends, spurious components and artifacts. The algorithms for preprocessing are usually the same for different signal types, the parameters (for instance: cut off frequency on pass-band filter) can be empirically tuned according to the specific application (for instance: band of noise).

**Extraction of information** aims at extracting from the acquired and preprocessed signal only the information of interest for the analysis. This operation could change the nature of the signal which can result in a different type respect to the input. For instance, we gather in this group algorithms like the extraction of Inter Beat Intervals (IBIs) from an Electrocardiogram (ECG), or the computation of the Phasic Skin Response (PSR) from Electrodermal activity (EDA). Since different signals also carry different information, these operations are usually specific for a specific type of signal.

**Computation of physiological indicators** aims at encoding key information from a signal into a set of scalar values. This stage is fundamental when diagnosing pathological states (for instance tachycardia can be observed from computation of mean heart rate) or for complex analysis

such as machine learning, when more than one metric (also called: feature) is computed and combined into a mathematical model to predict phenotypes. When interested in analyzing temporal variations, the indexes are computed on consecutive, and possibly overlapping, portions of the signal. This operation is known as windowing: windows used to segment the signal can have a fixed size or change according to the experimental conditions.

Alternatively, to the computation of indexes, the processed signals can be used as input to more complex analysis frameworks, such as Network Physiology [10] or Convolutional Neural Networks [92]. Although these analyses go beyond the purposes for which pyPhysio has been developed, some functions are provided.

## 3.2 Structure of pyPhysio

*pyPhysio* has been developed on top of well-known Python libraries for scientific analysis, in particular *scipy*<sup>1</sup>, *numpy*<sup>2</sup> and *spectrum*<sup>3</sup>.

In the following subsection we present the structure of the library: classes and their methods. In particular, we introduce the two novel classes: **Signal** describes general time-referred data and the class **Algorithm** which represents a processing procedure, with its subclasses **Filter**, **Estimator** and **Indicator**.

### 3.2.1 class Signal

From the computational point of view, a signal is an ordered vector of timestamp-value pairs, where the timestamp is the instant at which the measured phenomenon has that value. Both the timestamp and the value are approximates, due respectively to discrete nature of collected signals

---

<sup>1</sup><http://www.scipy.org/>

<sup>2</sup><http://www.numpy.org/>

<sup>3</sup><https://pypi.python.org/pypi/spectrum>

and quantization. In pyPhysio a signal is represented by the generic class `Signal`, containing the values of the signal and the information needed to reconstruct the temporal reference. An instance of `Signal` is associated to the measured physiological phenomenon by the attribute `type` which can be used to check whether signal-specific functions are erroneously applied. Usually the signal is sampled with a fixed sampling frequency, therefore it is sufficient to store the timestamp at which the acquisition started and the sampling frequency to reconstruct the timestamp of each sample. This type of signal is represented by the subclass `EvenlySignal`. An instance of class `EvenlySignal` stores the starting time of acquisition (as Unix time<sup>4</sup>) and the sampling frequency to recompute each sample timestamp when needed.

Other types of signals, for instance triggers indicating occurrences of heartbeats or events, are series of samples which are not equally temporally spaced. Thus the sampling frequency is not fixed and it is necessary to store the time reference of each sample. This type of signals is represented by the subclass `UnevenlySignal`. An instance of class `UnevenlySignal` stores the starting timestamp of acquisition (as Unix timestamp) and the vector of time references as elapsed time from the acquisition start.

Class `Signal` exposes default methods for manipulation of attributes:

- `get_values` and `get_times` return the vectors of values and of the timestamps of the signal respectively;
- `get_start_time` and `set_start_time` are used to get and set the timestamp of start of acquisition, for instance to synchronize signals acquired from different devices;
- `resample` is used to change the sampling frequency of the signal or to obtain an `EvenlySignal` from an `UnevenlySignal` instance.

---

<sup>4</sup>[https://en.wikipedia.org/wiki/Unix\\_time](https://en.wikipedia.org/wiki/Unix_time)

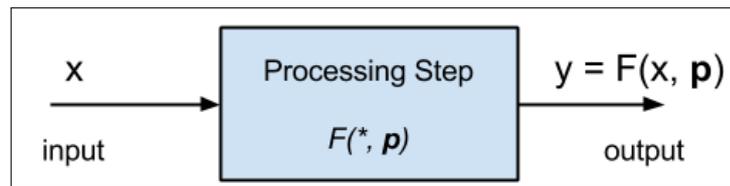


Figure 3.2: Representation of a generic algorithm characterized by the computational function  $F$  and its parameters  $\mathbf{p}$ .

The class `Signal` extends the `numpy.array` class of the Numpy package which is among the most important packages for scientific analysis in Python. Thus, each method developed in Numpy to work with a `numpy.array` instance will work also on a `Signal` instance.

### 3.2.2 class Algorithm

A signal processing algorithm is a computational function  $F$  that operates on input data (a signal) to produce a result. It is characterized by a set of parameters  $\mathbf{p}$  which regulate its behavior (see Figure 3.2).

In `pyPhysio` signal processing algorithms are represented by the new class `Algorithm`. This class is extended to represent each specific processing step by defining the computational function  $F$  and the set of parameters  $\mathbf{p}$ . To facilitate the creation of processing pipelines, in `pyPhysio` we kept separated three categories of algorithms: `Filters`, `Estimators` and `Indicators`, each one represented by its respective subclass. For a complete view of currently implemented algorithms see Table 3.2.

**Filter:** this subclass collects all algorithms for preprocessing (see first stage of the physiological signal processing pipeline, Figure 3.1). The type of the output signal from a `Filter` is the same of input signal (i.e. the attribute `type` is unchanged). Parameters can be set according to processing needs and characteristic of the signal.

**Estimator** subclass collects all the algorithms aiming at inferring key information from the input signal. As input signals are in general affected

by noise, having a discrete nature and only approximated descriptions of the real physiological phenomenon, we consider the extraction of information of interest as a stochastic process and resulting values as estimates. In addition, often the result can be strongly influenced by the value of algorithm parameters of which true values are unknown and are usually set by optimization or according to user expertise.

**Indicator:** this subclass collects the mathematical functions that compute a metrics from the input signal. Unlike **Filter** and **Estimator**, which return a signal, **Indicator** returns a single time-value pair, where the time is the starting time of the input signal. It is important to note that when the algorithm is executed within a windowing process the operation returns a list of time-value pairs that can be represented by an instance of class **Signal**. `pyPhysio` provides a collection of statistical, time- and frequency-domain and non linear metrics which can be adapted by appropriately choosing function parameters **p** to compute indicators for different types of signals. For instance, the algorithm to compute the power in a frequency band is the same for both EEG and HRV analysis, but the frequency bands are different and should be set accordingly (e.g. 8-12 Hz for alpha band of EEG, 0.14-0.4 Hz for High Frequency band of IBIs). `pyPhysio` provides default lists of indicators with specific parameters for each type of signal to compute main indicators proposed in literature (see Table 3.1).

### 3.3 Physiological signals

In the following paragraph we describe the main algorithms included in `pyPhysio`, grouped by type of physiological signal, emphasizing those aspects having a direct implication on psycho-physiological studies.

Signal	Indicators
Inter Beat Intervals	RRmean, RRSTD, RMSSD, pNN50, pNN25, pNN10, triang, TINN, VLF, LF, HF, LF/HF, SD1, SD2, SD12, Sell, ApEn, SampEn, DFAa1, DFAa2
Electrodermal Activity (PSR)	Mean, Range, Standard deviation, Mean peaks amplitude, Maximal peak amplitude, Peak slope, Peak duration, Number of peaks, AUC
Electroencephalogram	Energy in: delta, theta, alpha, sigma, beta wave bands
Electromyogram	Maximum, Minimum, Average, Range, SD, AUC, Energy in 4-40 Hz band
Respiration	Energy Low (0-0.25 Hz band), Energy High (0.25-5 Hz band), Energy Ratio, Breath Rate, Energy in 0-32 Hz bands
Activity	Maximum, Minimum, Average, Range, SD, AUC, Energy in 0-25 Hz bands

Table 3.1: Main physiological indicators provided by default in pyPhysio

Type	Name
<b>Filter</b>	Infinite Impulse Response filter, Adaptive Threshold, Matched filter, Convolutional filter
<b>Estimator</b> ECG/BVP EDA	Beat detection, Beat correction (see Appendix A) Optimization of Bateman parameters, Estimation of phasic and tonic components (see Appendix B)
<b>Tools</b>	FFT-based deconvolution, Power Spectrum Density estimation, Energy in band, Adaptive peak detection, Envelope estimation, Detection of local maxima and minima, Normalization methods, Matched filtering, Peak selection

Table 3.2: Algorithms provided in pyPhysio, grouped by type of signal

### 3.3.1 Cardiac signals

pyPhysio allows the processing of cardiac signals to extract HRV indicators, which are computed from the IBI signal. First step in the signal processing is therefore the detection of beats in the cardiac signal. pyPhysio provides two functions to estimate the position of beats in ECG and Blood Volume Pulse (BVP) signals respectively. The estimation of beat position in BVP signals is performed by the novel DBD-RCO algorithm that has been shown to improve the results in signals collected with WDs (introduced in Appendix A). The IBI signal can then be used to compute 20 HRV indicators.

### 3.3.2 Electrodermal Activity

pyPhysio provides two methods to decompose the EDA signal into its two components: the Tonic Skin Response (TSR) and the Phasic Skin Response (PSR). The first is the original algorithm proposed by Benedek [13], which is based on deconvolution using a Bateman function as Impulse Response Function. The second method is the peak-based algorithm introduced in Appendix B which aims at improving the estimation of Bateman parameters and identification of peaks. The estimated PSR signal is used to compute 9 indicators aiming at quantifying the amount of sympathetic activation.

### 3.3.3 Activity signals

pyPhysio provides algorithms to extract time and frequency indexes from signals acquired with standard inertial measurement units: accelerometer, gyroscope and magnetometer. As for other physiological signals these signals are first preprocessed in order to remove noise, then the indicators are computed. Since the physical observation is described by a vector in the

three-dimensional space, each sample is usually a tuple of three elements and usually an intermediate step is used to compute vector module. In the proposed pipeline, the derivative of each signal is also computed, then a set of time-domain and frequency domain indicators is computed on each signal and dimension.

### 3.4 Discussion

Designing experiments with multi-device settings is now a common practice in psycho-physiological studies as observing multiple physiological signals allows for better investigation of the activity of the autonomic nervous system.

Nowadays the technological progress is able to offer several devices for physiological data acquisition, from clinical instrumentation for in-lab settings to portable devices for flexible indoor studies up to wearable devices for real-life and in-the-wild applications. However, there is a gap between the increased availability of physiological sensing capabilities and the signal processing tools and software. This gap prevents research to fully exploit the amount of physiological information available. Often the main solution is to rely on commercial software which allows no customization of the algorithms and processing pipeline. As alternative solution the researcher has to deal with different libraries and programming languages (python, R, MATLAB) to be able to implement the desired analysis for each type of signal.

pyPhysio aims at filling this gap by providing a unique framework to work with different types of physiological signals. Beside implementing two novel algorithms for the analysis of BVP and EDA signals (see Appendixes A and B), it also provides a set of functions and algorithms that can be adapted to design processing pipelines for a broad set of physiolog-

ical signals. It is designed to require minimal knowledge of Python and programming skills: leveraging on the two main classes of the library, the creation of signal processing scripts is facilitated, as each step is represented by an object of the class `Algorithm` and all the over-head to manage and process a physiological signal is carried by the class `Signal`. Two examples are provided in Appendix ?? which demonstrate the usage of pyPhysio and how an HRV analysis pipeline can be easily implemented.

In addition, it has been conceived to facilitate the integration of new algorithms and the inclusion of more physiological signals such as Electro-oculography (EOG) and Magneto-encephalography (MEG). Finally, potential of pyPhysio can be fully exploited when integrated with other scientific Python libraries for downstream analysis (e.g. machine learning, scikit-learn, keras).

pyPhysio can be downloaded from <https://sites.google.com/site/pyhrvlib/> together with additional examples and tutorials.



## Chapter 4

# WCS: a dataset of Wearable and Clinical Signals for affective computing and reproducibility

The measurement of physiological signals is a fundamental process for the investigation of physiological phenomena and, from a psycho-physiological point of view, observation of autonomic regulatory mechanisms [68]. Being able to observe these phenomena in real-life and without constraints imposed by laboratory settings is a key reason for adopting WDs in scientific research [41]. However, beside many advantages, WDs have also relevant issues that affect the quality of acquired signals, including low sampling frequency, non-optimal positioning of the sensors and increased sensitivity to body movements (see Chapter 2).

Before WDs can be used in research, it is therefore fundamental to verify that results achieved are comparable with those obtained from signals acquired by medical-grade devices. In particular, signal processing algorithms are expected to give comparable results considering the required level of precision according to the type and purpose of application. However, despite its importance, this comparison is often omitted or left to manufacturers who rarely provide examples dataset for validation.

Several datasets with physiological signals are available, such as DEAP [67], MANHOB-HCI [114] and SEMAINE [77]. A rich archive of physiological signals is also available through PhysioBank [49]. However, to validate WDs we need a dataset comprising both types of devices: medical-grade to provide reference signals and wearable for comparison. As none of the existing resources satisfies this requirement, we developed the Wearable and Clinical Signals (WCS) dataset which is introduced in this Chapter. In Section 4.1 we present the experimental settings and provide details about the devices composing the sensing architecture. Section 4.2 is devoted to describe the methodology adopted to synchronize the signals, while in Section 4.3 we provide the details of the experimental design. Information about the resulting dataset and discussion is presented in Section 4.4 and 4.5.

## 4.1 Devices and architecture

To be able to assess reproducibility of studies based on WDs, we need a dataset including signals from both types of devices: medical- and wearable-grade. In this section we present the experimental architecture adopted to collect synchronized signals and create the Wearable and Clinical Signals (WCS) dataset. We describe the types of signal acquired, sensing specifications and settings for each device.

### 4.1.1 Selection of WDs

The high number of WDs currently available would make unfeasible to extensively validate all existing solutions, therefore we decided to start focusing on high performance devices. Inadequate results obtained from high performance WDs will consequently arise doubts about the possibility of using WDs with even lower performance for reproducible studies.

However, as the landscape of available WDs solutions is highly heterogeneous, some requirements need to be imposed to select the most appropriate devices for the investigation. In particular, we focused on identifying the key characteristics of WDs to allow usage in everyday life. We defined four criteria to identify the candidate WDs to be used in the experiment:

1. **Wearability:** the device should be easily worn by the subject with no need of applying conductive gels, wired electrodes or similar preparations. In fact, in everyday applications the presence of an external supervisor over-viewing the preparation (e.g. experimenter) is not expected;
2. **Streaming:** the device should stream the data in real-time to an external collector through Bluetooth connection. The Data Streaming Protocol (DSP) should be provided by the manufacturer. Streaming capability through available DSP is a key feature which allows real-time processing and feedback, for instance used in remote health monitoring applications;
3. **Availability:** the device should be commercially available at the moment of testing (prototypes, proof-of-concepts or custom devices are not considered). This requirement is imposed to restrict the selection to state-of-art technologies, which could be accessed by researchers with any technical and/or technological competences;
4. **Performances:** the final choice should favour the device with higher sampling rate and higher number of sensors to select the most advanced technological solution.

After a preliminary phase in which a list of WDs (see Appendix C) was considered to test compliance with defined criteria, we identified two devices: the Empatica E4 and the Comftech HeartBand. Empatica E4 was

chosen as a representative of wristband devices, while Comftech HeartBand belongs to the smart-garments category.

#### 4.1.2 The baseline: Thought Technology FlexComp

The clinical device we used as reference is the Thought Technology FlexComp unit (commercial code: T7555M). It is a customizable acquisition unit which provides up to 10 input slots that can be used to connect diverse physiological sensors. Maximal sampling frequency is 2048 Hz, which is the sampling frequency adopted for the experiment, with 14 bits of resolution for each input. Signals acquired for the experiment are:

1. Electrocardiogram (ECG): using three electrodes placed over the left and right coracoid processes and below the ribs on the left. The signal is pre-amplified and filtered by the EKG Sensor (T9306M) which returns a single channel read in millivolts. UniGel electrodes (T3425) are used as conductive mean between the sensor and the skin;
2. Electrodermal Activity (EDA): two finger bands with Ag-AgCl electrodes (SA2659) are placed on the second and fourth finger of the left hand and connected to the sensor (SA9309M). The skin conductance is measured in microSiemens ( $\mu\text{S}$ );
3. Blood Volume Pulse (BVP): the sensor (SA9308M) is placed on the third finger of the left hand. The relative amount of reflected infrared light is measured;
4. Respiration (RESP): a band is worn on the chest to measures relative volumetric expansion by elongation of an elastic patch (SA9311M);
5. Trigger (TRG): a handle with a button to generate electrical impulses used to manually mark the experimental events.

Setup with positioning of the sensors was supervised by a trained experimenter to ensure high quality signals. The acquisition unit is connected through USB to a personal computer (running Windows 10) where the proprietary BioGraph Infinity Software Platform manages the acquisition process and collects the signals.

### 4.1.3 Empatica E4

The Empatica E4 [44] is a multi-sensor wristband designed for real-life acquisitions of physiological signals. It has streaming (through Bluetooth Low Energy) and storing (internal flash memory) capacity; communication with an Android smartphone is provided by EmpaLink SDK. Collected signals are:

1. BVP: four Light Emitting Diodes (LEDs) are used to generate light at two different wavelengths (Green and Red) and two photodiodes are used to measure reflected light. Using two wavelengths and an appropriate proprietary algorithm to preprocess the signals should reduce motion effects and sensitivity to external sources of light. Sensors are placed on the bottom of the wristband case in firm contact to the skin; signal is sampled at 64 Hz;
2. EDA: two stainless steel electrodes are placed on the band to allow positioning on the inner side of the wrist. Skin conductance is measured in microSiemens at 4 Hz sampling rate;
3. Acceleration (ACC): three axes acceleration (range  $\pm 2g$ ) is measured at 32 Hz sampling frequency;
4. Skin Temperature (ST): measured by an infrared thermopile placed on the back of the case, at 4 Hz sampling frequency.

Placement of the wristband was supervised by the experimenter to ensure the device is not worn too tight (which would prevent physiological blood flow) or too loose (which would prevent the smartwatch case to be in contact with the skin) to guarantee optimal conditions for a correct BVP signal acquisition. However, in line with the idea that WDs are to be used in real-life context, where the presence of an experimenter is not expected, each participant was allowed to minimally adjust the positioning to favor comfort during the experiment.

#### 4.1.4 ComfTech HeartBand

The Comftech HeartBand is an elastic band with embedded tissue electrodes and an acquisition unit that is connected to the band through two snaps. Streaming is based on Bluetooth 2.0 and the documentation to decode the hexadecimal messages from the device was provided by the manufacturer.

Collected signals are:

1. ECG: using two tissue electrodes placed over the chest and connected to the acquisition unit where the signal is sampled (128 Hz), pre-amplified and filtered to return a single channel read;
2. ACC: three axes acceleration is measured by a sensor embedded on the acquisition unit; sampled at 200 Hz;

As for the Empatica E4, the participant was allowed to adjust the positioning of the band, after having received detailed instructions by the experimenter. The quality of the acquired signals was checked by visual inspection before starting the experiment, in particular verifying that the ECG signal exhibits patterns associated to the QRS complex.

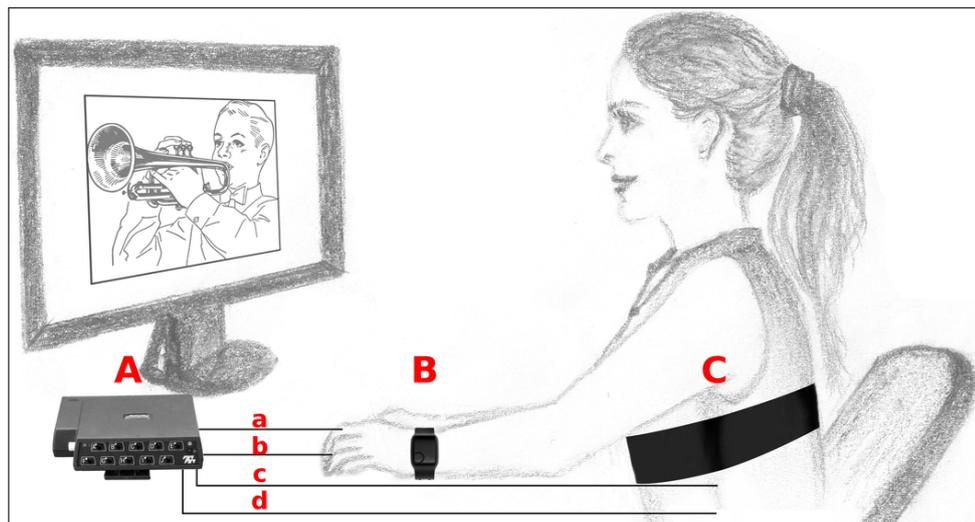


Figure 4.1: Illustration showing the devices used for the acquisition of physiological signals. A: Thought Technology FlexComp (picture from [www.thoughttechnology.com](http://www.thoughttechnology.com)) with four wired sensors: a) Electrodermal activity, b) Blood Volume Pulse, c) Electrocardiogram, d) Respiration; B: Empatica E4 (picture from [www.empatica.com](http://www.empatica.com)); C: ComfTech HeartBand.

### 4.1.5 PhysioREC

The PhysioREC app (see 2.2.1) was installed on a Samsung Tab A running Android 5.1 to manage signal acquisition from the two WDs. Two input plugins to communicate with the Empatica E4 and the ComfTech HeartBand respectively were enabled; an additional input (MRK) signal can be generated by a button on the PhysioREC User Interface to mark experimental events. Before starting the experiment, the signals acquired through the two WDs are visually inspected by activating the real-time plotting function. Acquired signals are sent to Physiokitix database, where an automatic process takes care of indexing and storing with correct meta-data.

## 4.2 Synchronization

Synchronization is crucial in multi-device experiment settings. In this case, two flows of physiological signals can be identified: the medical-grade data, from the FlexComp unit to the personal computer, and the wearable-grade data, from the WDs to the tablet and then to the Physiokitix database. Unfortunately, the two flows are independent both in terms of path of the data, from generation to storage, and time reference. To synchronize, we exploited the two manual inputs provided in both flows, the TRG signal in the FlexComp and the MRK on the PhysioREC user interface. The experimenter pressed both simultaneously at the beginning and end of the experimental session, thus providing two manual reference points to allow synchronization during the signal processing stage.

However, this methodology is not able to solve the synchronization issues originated by latency in WDs. A more precise synchronization is then performed by computing the cross correlation function between pairs of Inter Beat Intervals (IBIs) signals.

We extracted the IBIs from the cardiac signal of each device (ECG for FlexComp and HeartBand, BVP for Empatica) thus obtaining three IBIs: the reference from FlexComp and the remaining IBIs (from E4 and from HeartBand) to be synchronized by pair-wise comparison with the reference. We randomly selected on the baseline portion 20 segments (length: 60 seconds) and for each segment we computed the cross correlation function between reference IBI and the desynchronized IBI (from the WD). Local temporal offset was identified as the instant at which the cross correlation function is maximal. The global temporal offset between each pair was estimated by averaging the offsets across all segments and used to synchronize the signals from each WD.

Session	Duration [mins]
Instructions and setup	
<b>Start of acquisition</b>	
Baseline	5
<i>Pause</i>	2
Stimulation	50
<i>Pause</i>	2
Moving	5
<b>Stop of acquisition</b>	

Table 4.1: Sessions of the experiment to create the WCS dataset.

### 4.3 Experimental phases

The experiment comprises three phases: baseline, stimulation and moving. Each phase has been conceived with specific goals which are described in the following subsections. An overall description of the sequence of experimental phases is presented in Table 4.1.

#### 4.3.1 Baseline and moving

First and last phases are mainly focused on providing signals to test processing algorithms.

During baseline the participants are asked to remain sit and still. Signal collected during this phase represent a reference for the maximal signal quality that can be provided by WDs, as all external sources of noise that could affect the signal quality (such as moving artifacts, sensor displacement or detachment) are experimentally avoided. This condition is far from the real-life context in which the WDs are expected to be used, but it is needed to isolate effects of technical limitations and constraints from other causes of errors.

Effects of body movements (and technical limitations and constraints) can be evaluated on the moving phase which concludes the experiment. During this phase the subject is asked to stand and simulate a walking

on place, as the free movement was prevented by the wires connecting the medical-grade sensors with the FlexComp unit. The experimenter suggested moving naturally to replicate the intensity of usual walking. In particular, special attention was paid to movements of the left hand as, due to presence of sensors, participants tended to keep it still and near to the body. This experimental phase is expected to be more similar to real-life contexts and can be used to evaluate performances of algorithms for every-day applications, such as health monitoring, fitness and stress detection.

### 4.3.2 Stimulation

Affective computing is one of the main fields of research where acquisition of physiological signals by WDs is expected to give a major contribution. In fact the suggested use cases of commercial WDs are mostly stress monitoring, relaxation, control of anxiety and other applications for the management of emotions. In addition, affective computing naturally benefits from multi-modal experimental setups and thus allows to fully exploit the potential of WDs with multiple sensors.

In particular, the possibility of monitoring emotions in real-life context could enhance efficacy of therapy of Autism Spectrum Disorders [21, 34].

For this reason, in the stimulation phase we replicated the experimental design proposed in [67] to create the DEAP dataset. In the original study 32 subjects watched 40 music-videos stimuli of 60 seconds. The composition of the list of stimuli was based on an on-line survey which initially considered 120 videos from which the 40 with stronger emotional content were finally selected. The selection of the 60 second length segment of each video was performed automatically by application of an algorithm derived from [113] which identifies the portions with higher emotional contents, based on audio and video features.

After each stimulus, participants rated the emotional content according to five dimensions: valence, arousal, dominance, liking and familiarity on a 1 to 9 continuous scale. This dataset is suited for application of algorithms for affective computing where the physiological signals can be used to predict the emotional content of the video.

The choice of replicating the DEAP experiment was driven by the following motivations:

1. The DEAP dataset is well known in the field of affective computing and has been adopted in several emotion recognition studies. This provides a rich state-of-the-art background to which our study about reproducibility of WDs can compare;
2. The stimuli adopted in the DEAP dataset can be easily retrieved: this aspect facilitated the preparation of the experiment but could also motivate the replication of the same study with new WDs, thus allowing the expansion of the dataset.

For the same reason, we decided to skip the rating of the videos such as the software required to present the stimuli is simplified and corresponds to JavaScript-based web-page that can be run by a simple browser without need of further software. The web app retrieves the list of video files contained in an input folder, randomize the order and presents the videos with 15 seconds of pause between each video. At the end of the experiment it generates a log file with the name, start and stop timestamps of each video.

The DEAP dataset includes also information about the list of links to YouTube videos used to create the stimuli, together with the instant at which the 60 seconds segment begins. Based on this information we were able to recreate the set of original stimuli. However, we were unable to retrieve 14 videos, as they have been removed from the YouTube archive. We

substituted the missing videos with other videos following the indication about the Artist and the name of the song. Videos have been retrieved with the highest available video and audio quality and rescaled to fit an 800x600 pixel frame. Videos which had a different aspect ratio were forced to have maximal width of 800 pixels, while automatically adjusting the height to preserve the original aspect ratio.

All the data collection experiment was developed in collaboration with researchers and students of the Affiliative Behaviour and Physiology (ABP) Lab, University of Trento; recording took place in the ABP Lab in Rovereto (Trento) from September to December 2016.

## 4.4 Results

A total of 18 participants were recruited for WCS; each participant signed the informed consent and the experiment was conducted according to the principles of the Declaration of Helsinki [128]. The data have been anonymized; further we transformed the timestamps from absolute to relative, in order to prevent any possibility of recognizing the identity of participants.

Signals have been preprocessed with the following pipeline:

1. Synchronization: alignment of the TRG and MRK position as first, then precise synchronization by cross-correlation to correct latency;
2. Segmentation: each phase of experiment was segmented in order to facilitate loading of data and remove pauses;
3. Re-sampling: signals were resampled to 1024 Hz. Oversampling was performed by cubic-spline interpolation;
4. Saving: a table with all signals was created, compressed and saved.

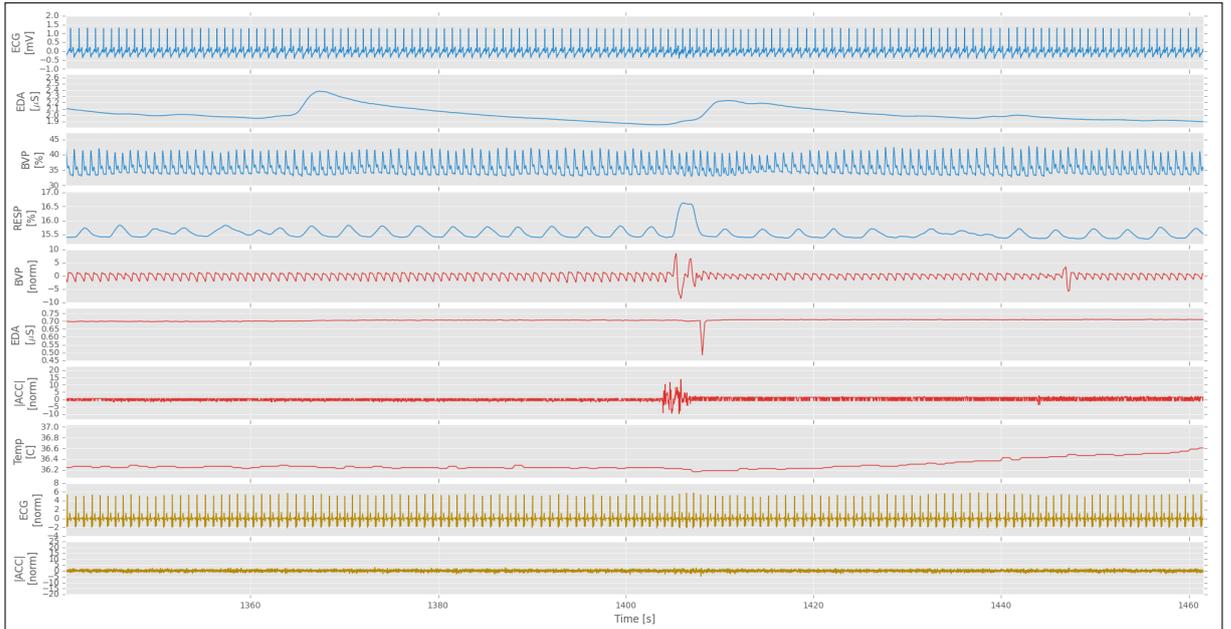


Figure 4.2: A 120 seconds length segment of WCS signals. Blue: signals collected with medical-grade device (FlexComp); Red: signals collected with Empatica E4; Yellow: signals collected with ComfTech HeartBand. Note effects of body movements on the BVP signal of Empatica E4 due to a sudden chest movement (probably a cough) observable in the central part of the plot (respiration signal) and on the module of Acceleration from Empatica E4.

The final size of the WCS dataset is approximately 2.8 Gb. An example of acquired signals can be found in Figure 4.2.

## 4.5 Discussion

In this chapter we presented the WCS dataset we created to compare medical-grade and wearable devices. Three phases have been conceived in order to allow testing of algorithm for different purposes and contexts. Baseline phase should provide a benchmark to test reproducibility of existing algorithms on signals from WDs in the ideal case, while the moving phase is more oriented to provide a realistic data for real-life applications. The stimulation part aims at reproducing the DEAP dataset which is a well known dataset in the field of affective computing, thus allowing testing

reproducibility of emotion recognition experiments using WDs.

As testing a broad number of WDs would be unfeasible, we decided to focus on high performance WDs: Empatica E4 and ComfTech HeartBand, to represent the classes of wristband devices and smart garments respectively, which can be considered as the gold-standard of WDs. However, the experimental design was kept as simple as possible to encourage contributions from other researchers to expand the dataset with novel WDs.

As we discussed in Chapter 2, many experimental factors and technical differences can contribute to change considerably the final quality of signals acquired with WDs. So far, each study presented its own results based on a custom dataset appositely created. Instead, to allow comparison between different studies and test reproducibility, we propose the adoption of the WCS dataset as a standard benchmark to which the validation of new algorithms for signals from WDs should refer. Beside, we expect manufacturers to refer to the WCS dataset and its experimental design to show improvements of new technologies.

## Chapter 5

# Assessing reproducibility of wearable devices

One of the main fields in which WDs are expected to add a major contribution is the automatic recognition of affective states by physiological signals, also defined by Rosalind Picard, a pioneer in this field, “affective computing” [88]. Solving this task would be relevant in many diverse disciplines areas, such as the development of Human Computer Interfaces (HCI) [43], the treatment of psychological pathologies [91] and the development of new technologies to train parents to deal with children with special needs [106]. Experimental settings with multiple signal are adopted in several psychophysiological studies aimed at investigating the regulation by the ANS. In particular, a multi-modal analysis has been adopted to understand how the human body adapts to different psycho-cognitive states [80] or when exposed to external stimuli or stress [86, 107].

Affective computing from physiological signals is now gaining interest, due to the increasing availability of WDs, which would allow a more extensive acquisition of physiological signals also in real-life applications. This aspect paves the way to two key improvements in affective computing: (a) the possibility of designing experiments in ecological setups, for instance domestic contexts, where human behaviour and physiology is expected to

be more natural; (b) the application of emotion recognition techniques for therapeutic purposes, for instance to treat psychological and cognitive disorders. Affective computing experiments are commonly based on a multi-modal setup, which is naturally provided by the WDs that embed multiple sensors such as the Empatica E4. Finally, the affective computing field is supported by a solid background of scientific research which gives the possibility for validation of experiments. These are the main reasons we chose the affective computing field to assess whether WDs are able to reproduce state-of-art results obtained with medical-grade devices.

In particular, we focus on the DEAP dataset and on the stimulation part of the WCS dataset (see Chapter 4) to compare results on an emotion recognition task. However, before investigating the performance of the emotion recognition task, the reliability and reproducibility of the algorithms adopted in the physiological signal processing pipelines (see 5.1.2) have been evaluated. Results are presented in the Appendixes A and B, for the analysis of cardiac signals and EDA respectively.

It is worth noting that classification of emotions based on the DEAP dataset is a difficult task as demonstrated by previous studies (see a summary Table 5.1). These studies will be used as references to test and validate the proposed approach. Koelstra and colleagues [67] focused on the classification of high and low levels of arousal, valence and liking. Three different subject-dependent classifiers were defined (one for each dimension). The labeling of the samples was performed by thresholding the subject's ratings (threshold value is the middle of the 1 to 9 rating range). 106 features were extracted from the 7 peripheral signals and a naive-bayes classifier with a leave-one-out cross validation scheme was adopted. The average accuracy over all subjects was: 57% for arousal, 62.7% for valence, 59.1% for liking. Note that expected accuracy in case all samples were assigned to the more represented class in the dataset, would be respectively:

Reference	Type of classifier	Subject dependent	Type of validation	Signals	Arousal	Valence
Koelstra 2012 [67]	Naive-Bayes	yes	Leave-one-out	BVP, EDA, RESP, ST, EMGz, EMGt	57.0	62.7
Chung 2012 [31]	Bayesian	yes	Not available	EEG	66.4	66.6
Torres 2014 [122]	HMM	no	10x5	BVP, EDA, RESP, ST	55.0	57.5

Table 5.1: Summary of main state-of-art accuracies for the recognition of arousal and valence based on the DEAP dataset. BVP: Blood volume pulse; EDA: Electrodermal activity; RESP: Respiration; ST: Skin temperature; EMG: Electromyogram, EMGz: EMG of the Zygomaticus muscle, EMGt: EMG of the Trapezius muscle; EEG: Electroencephalogram; HMM: Hidden Markov Model

64.4%, 58.6%, 67.0%. Chung and colleagues [31] focused on the classification of high and low levels of arousal and valence (labeled as in [67] considering the EEG signal from which 392 spectral features were extracted. A subject-dependent bayesian classifier was adopted. Resulting average accuracies across all subjects were 66.4% for arousal and 66.6% for valence. An additional multi-class test was performed assigning the labels to the class high (subject rating greater or equal than 6), low (subject rating less than 4) or to medium (the remainders). Average accuracies across all subjects for the second test were 53.4% for arousal and 51% for valence. Torres and colleagues [122] proposed to develop a subject-independent classifier for arousal and valence based on Hidden Markov Model with a 10x5 cross validation scheme. Samples were assigned to the class high or low class as in Koelstra et al. [67]. Different datasets were obtained by differently combining the features from different physiological signals. The accuracies when considering only the features extracted from the BVP, EDA, skin temperature and respiration were 55% for arousal and 57.5% for valence.

In this Chapter we first present the methodology adopted for validation (Section 5.1), in particular: signal processing and machine learning pipelines and classification metrics, then presents results (Section 5.2) of comparisons between different datasets and pipelines. In Section 5.3 we summarize and comment the findings.

## 5.1 Modeling and validation

The main goal of this analysis is to assess whether WDs are able to provide results comparable with experimental settings based on medical-grade devices. To this aim we focused on a typical affective computing task: recognizing the emotional content of a stimulus based on indicators computed from physiological signals.

Two dataset were considered: the DEAP dataset, which is the state-of-art reference and the WCS which reproduces the experimental setup of the DEAP dataset also including signals from WDs (see Chapter 4). In both cases the objective was to recognize the emotional content of music videos used as stimuli.

In summary, the analysis was performed on the following sets of signals:

1. Signals provided in the DEAP dataset (DEAP), comprising: BVP, EDA;
2. Signals collected by the medical-grade device FlexComp (WCS-Medical), comprising: ECG, BVP, EDA;
3. Signals collected by the WD Empatica E4 (WCS-E4), comprising: PPG, EDA;
4. Signals collected by the WD ComfTech HeartBand (WCS-HB), comprising: ECG.

The DEAP dataset was first used as reference to validate the algorithms proposed for signal processing, then to assess whether our experimental design reproduces original design proposed to create the DEAP dataset. Based on the WCS-Medical we will validate the two WDs used to compose the WCS-E4 (Empatica E4) and the WCS-HB (ComfTech HeartBand).

In the following subsections we provide details of the methodological approaches adopted for validation.

### 5.1.1 Definition of problem

Final goal of the analysis is to use machine learning (ML) predictive models to recognize the emotional content of each music video stimulus based on indicators computed from the physiological signals acquired with the medical-grade and wearable devices. In the DEAP dataset, together with the physiological signals, subjective ratings for each stimulus were also provided, collected by the participants and by an on-line survey. Ratings were given according to four emotional dimensions: Arousal, Valence, Liking and Dominance.

In order to compare results from the DEAP and the WCS dataset, which does not include subjective ratings, we trained the ML models using labels obtained from the on-line ratings of the DEAP dataset, focusing on Arousal and Valence dimensions. Each stimulus is labeled as belonging to the high or low class of each dimension depending on whether the on-line rating is above or below 5, which is the middle value of the 1-9 likert scale used for rating.

### 5.1.2 Signal processing pipelines

In the analysis we concentrated on cardiac signals (ECG and BVP) and EDA only, as only these signals are available in both versions: from medi-

cal-grade and wearable devices. Adopted processing pipelines differ for each signal:

- BVP signals: we compared our novel DBD-RCO algorithm (DBD) (see Appendix A) with the algorithm proposed by Aboy and colleagues [1] (ABOY);
- ECG signals: we adopted a standard algorithm based on adaptive peak detection (APK);
- EDA signals: we compared our novel algorithms for optimization of Bateman parameters and estimation of the Phasic Skin Response (PSR) component (see Appendix B) (PKB) with original algorithms from Benedek and colleagues (BEN) [13].

It is worth noting that the quality of EDA signal obtained from the Empatica E4 was poor in this setup and thus we were unable to perform analysis. In addition, Subject 14 had a sensor disconnection during the stimulation part and therefore it was thus dropped out.

A signal processing pipeline was applied to compute a vector of indicators to be used as input features of the following ML pipeline. In details, the signal is segmented into 40 portions corresponding to the 40 music-video stimuli; each segment is used to extract the physiological indicators associated to the signal and labeled according to the emotional content of the stimulus. Heart Rate Variability (HRV) indicators have been computed from cardiac signals and tonic and PSR metrics from EDA signals.

An overall view of the results of the signal processing pipeline on each set is provided in Table 5.2. Signal processing was performed with custom Python scripts based on pyPhysio (see Chapter 3).

Source	Signals	Algorithms	Number of subjects
DEAP	BVP	ABOY	32
		DBD	32
	BVP+EDA	ABOY+BEN	32
		DBD+PKB	32
WCS-Medical	BVP	ABOY	17
		DBD	17
	BVP+EDA	ABOY+BEN	17
		DBD+PKB	17
WCS-E4	BVP	ABOY	17
		DBD	17
WCS-HB	ECG	APK	17

Table 5.2: Source datasets, physiological signals and processing pipeline applied: ABOY [1], DBD (see Appendix A), BEN [13], PKB (see Appendix B), APK: adaptive peak detection

### 5.1.3 Machine Learning pipeline

Regarding the ML pipeline, we adopted a Support Vector Machine (SVM) model with linear kernel as classification model and the MAQC-III/SEQC data analysis pipeline [110] for unbiased feature selection and classification (see Figure 5.1). The pipeline was initially proposed to develop predictive models from gene expression data [32] and was selected as it represents a validated reference for reproducible and reliable machine learning studies. The SVM model was selected to reproduce the framework proposed in the reference work of Torres and colleagues [122].

In its general form, the ML pipeline is composed of three nested cycles. The first cycle corresponds to the Cross Validation (CV) in which the dataset is split into two parts according to chosen scheme: one (training set) is used to train the predictive model, the other (validation set) to validate the trained model and obtain classification performance. The procedure is iterated over all parts and usually is repeated ten times (see

Figure 5.1, step a).

The second internal cycle tunes model parameters. The training set is split in half: its first part is used to tune the regularization parameter  $C$  and the second is used to test performances with that value. The procedure is repeated ten times (10x2) in order to select optimal value (see Figure 5.1, step b).

The third step implements a feature selection procedure: features are sorted (see Figure 5.1, step c) according to a given rank (computed internally or manually defined) then starting from top-ranked features a new model is defined, with increasing number of features (see Figure 5.1, step d). This procedure attempts to improve the classification results by automatically identifying the optimal set of features and rejecting those less discriminant and, possibly, noisy.

Overall performance is then computed over the different models (one for each subject), i.e. the classification metrics are averaged across all subjects

Two different ML pipelines have been derived depending on the CV scheme and method to rank features.

#### **ML pipeline 1: LOO CV ML pipeline**

First we reproduced the same scheme proposed in [67], which uses a Leave-One-sample-Out (LOO) CV. In this scheme the train-test cycle is iterated 40 times for each subject: each time a different sample is left out from the training set to compose the validation set. Features are ranked internally using the weights of the SVM model. Classification performances are computed at the end of 40<sup>th</sup> cycle when all samples have been predicted in the validation.

However, this ML pipeline is not appropriate to compare different datasets and pipelines for two key reasons:

1. The LOO CV scheme is prone over-fitting and thus it would provide

a biased evaluation of classification performance [24];

2. As the feature selection procedure is internal to the modeling pipeline, the feature ranking will depend on the signal quality and processing pipeline of each dataset, thus providing biased results.

### **ML pipeline 2: 10x5 CV with fixed feature ranking**

To solve these problems, we defined a ML pipeline adopting a different CV scheme and a fixed feature ranking. We choose a 10x5 CV scheme, where the dataset is split into 5 parts. Each part is iteratively selected to compose the validation dataset, while remaining parts are used to train the model. The whole procedure is repeated 10 times, each time randomly shuffling the samples composing each part.

To obtain the fixed feature ranking we adopted the following procedure. First, we trained a predictive model for each subject of the DEAP dataset (DBD-PKB pipeline) using a 10x5 CV scheme with internal feature ranking based on SVM weights, thus obtaining the feature ranking for each subject. Then we computed the unified feature ranking by merging all feature rankings using the Borda algorithm [65, 63]. Resulting list is used as the fixed feature ranking which is provided to the ML pipeline. A fixed feature ranking was obtained for both dimensions: Arousal and Valence.

#### **5.1.4 Classification performances**

To validate the processing pipelines and assess whether WDs reproduce results of medical-grade devices we compared the classification performances of models from the different datasets. Three metrics are reported: the Matthews Correlation Coefficient (MCC) [64], F1-score (F1) and Accuracy (ACC). The first has been chosen to better describe the results of the classification, the second to compare with the results reported in [67],

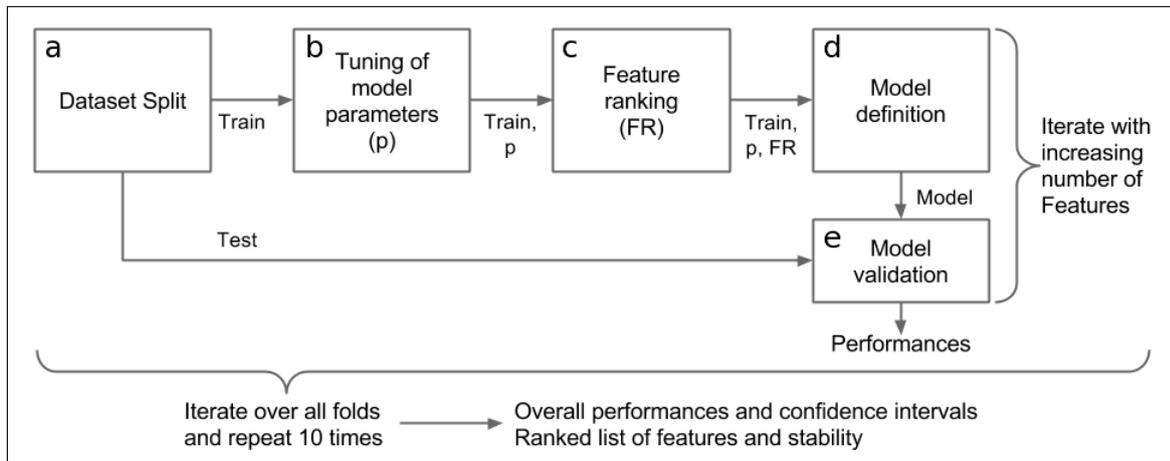


Figure 5.1: The ML pipeline for unbiased classification proposed within the SEQC/MAQC-III consortium [110]. In (a) the input dataset is split into 5 subsets (folds), maintaining the class stratification. A fold is kept for test, the remaining are used to train the model in the following steps. In b) a first predictor is used to optimize the parameters of the model by grid-search. An internal CV (e.g. 10x2-CV) is used to avoid overfitting. The tuned classifier is used to estimate the feature ranking (c). A second predictive model is defined (d), taking in input a reduced set of features (selected by previous ranking). Its performance is evaluated on the test set (e). The step is iterated considering an increasing number of features. The whole procedure is performed over all the folds and repeated 10 times to evaluate the overall performance of the model (e.g. MCC, Accuracy), the overall feature ranking and its stability over the iterations.

while Accuracy is the most adopted metrics to report classification results. For each model, classification metrics are obtained from the feature step where the MCC is maximal. In case of 10x5 CV pipeline, bootstrapped 95% confidence intervals (CI) are also computed from the empirical bootstrap distribution of the average MCC. As models are defined for each subject, overall classification performances were computed as average and bootstrapped 95% CI across all subjects. All software was implemented in Python using custom scripts based on *scikit-learn* and *mipy* packages for ML.

## 5.2 Results

In this section we compare the classification performance between the different datasets and the adopted pipelines to assess whether WDs can be used to reproduce scientific results in emotion recognition (see Figure 5.2).

We first validated the methodological approach (signal processing pipeline and ML pipeline), then the WCS dataset and, finally, the two WDs: Empatica E4 and ComfTech HeartBand.

### 5.2.1 Comparison with background studies

As first, we assessed whether our methodology is able to provide results which are comparable with background studies proposed in literature. When using the LOO CV scheme (ML pipeline 1), we improved the results achieved in [67], for both Arousal and Valence dimensions (see Table 5.3).

Further, to tackle the issues with the LOO CV pipeline described in the original paper from Koelstra and colleagues [67], we consider the results from the second ML pipeline: with 10x5 CV scheme and fixed feature ranking. In this case we compare our results with those presented in the

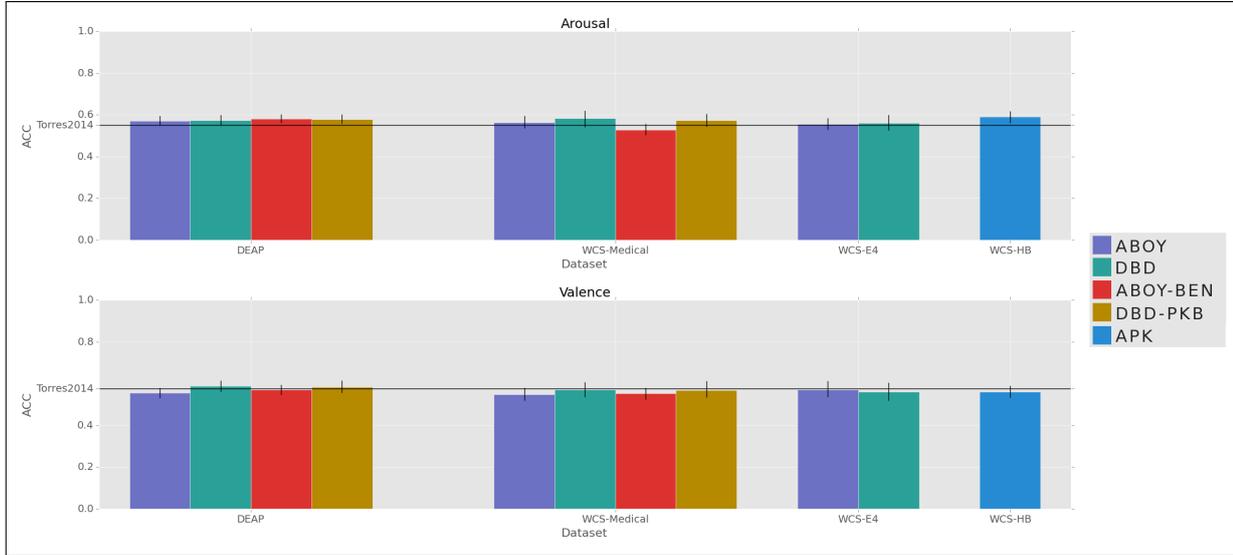


Figure 5.2: Overall accuracy on all datasets. 95% Confidence intervals computed by Student's bootstrapping.

		<b>Arousal</b>				
		Koelstra2012[67]	ABOY	DBD	ABOY-BEN	DBD-PKB
MCC	Not available		0.576 <i>(0.470-0.673)</i>	0.672 <i>(0.585-0.770)</i>	0.576 <i>(0.494-0.660)</i>	0.752 <i>(0.657-0.844)</i>
F1	0.533		0.778 <i>(0.717-0.833)</i>	0.84 <i>(0.797-0.886)</i>	0.769 <i>(0.720-0.815)</i>	0.877 <i>(0.826-0.925)</i>
ACC	0.570		0.783 <i>(0.728-0.832)</i>	0.834 <i>(0.790-0.883)</i>	0.781 <i>(0.741-0.823)</i>	0.873 <i>(0.826-0.920)</i>
		<b>Valence</b>				
		Koelstra2012[67]	ABOY	DBD	ABOY-BEN	DBD-PKB
MCC	Not available		0.626 <i>(0.509-0.735)</i>	0.606 <i>(0.503-0.705)</i>	0.709 <i>(0.616-0.798)</i>	0.615 <i>(0.507-0.716)</i>
F1	0.608		0.808 <i>(0.747-0.863)</i>	0.793 <i>(0.739-0.846)</i>	0.855 <i>(0.812-0.900)</i>	0.799 <i>(0.745-0.851)</i>
ACC	0.627		0.809 <i>(0.749-0.864)</i>	0.799 <i>(0.747-0.849)</i>	0.852 <i>(0.805-0.896)</i>	0.805 <i>(0.751-0.855)</i>

Table 5.3: Classification performances achieved on the DEAP dataset with LOO CV scheme according to different pipelines.

		<b>Arousal</b>				
		Torres2014[122]	ABOY	DBD	ABOY-BEN	DBD-PKB
MCC	N. A.		0.155 <i>(0.108-0.203)</i>	0.165 <i>(0.114-0.215)</i>	0.177 <i>(0.135-0.218)</i>	0.173 <i>(0.132-0.216)</i>
F1	N. A.		0.581 <i>(0.561-0.602)</i>	0.569 <i>(0.542-0.596)</i>	0.576 <i>(0.559-0.594)</i>	0.572 <i>(0.548-0.595)</i>
ACC	0.55 $\pm 0.039$		0.571 <i>(0.549-0.593)</i>	0.574 <i>(0.551-0.598)</i>	0.581 <i>(0.562-0.600)</i>	0.579 <i>(0.561-0.600)</i>
		<b>Valence</b>				
		Torres2014[122]	ABOY	DBD	ABOY-BEN	DBD-PKB
MCC	N. A.		0.124 <i>(0.072-0.174)</i>	0.192 <i>(0.132-0.248)</i>	0.16 <i>(0.109-0.208)</i>	0.187 <i>(0.129-0.246)</i>
F1	N. A.		0.556 <i>(0.531-0.578)</i>	0.587 <i>(0.555-0.616)</i>	0.582 <i>(0.555-0.607)</i>	0.597 <i>(0.570-0.622)</i>
ACC	0.575 $\pm 0.039$		0.556 <i>(0.532-0.579)</i>	0.588 <i>(0.562-0.614)</i>	0.571 <i>(0.547-0.594)</i>	0.584 <i>(0.558-0.612)</i>

Table 5.4: Classification performances achieved on the DEAP dataset with 10x5 CV scheme according to different pipelines.

study of Torres and colleagues [122] which also uses a 10x5 CV scheme with Hidden Markov Model for classification (see Table 5.4). As expected, in this case the overall performance decreases. However, in case of Arousal, all pipelines appear to improve results from Torres and colleagues [122] with no evident difference between the pipelines. Datasets composed of indicators of HRV and of tonic and phasic skin response (ABOY-BEN and DBD-PKB) achieve a slightly better performance. In case of Valence, only pipelines based on the novel algorithms appear to achieve, and fairly improve, results from Torres and colleagues [122]. It is worth mentioning that they used two more sources as input signals (Respiration and Skin temperature) which have not been considered in this work. This preliminary analysis confirms that proposed methodology is in line with state-of-art of emotion recognition/affective computing.

<b>Arousal</b>				
	ABOY	DBD	ABOY-BNDK	DBD-PKB
MCC	0.138 <i>(0.080-0.203)</i>	0.177 <i>(0.093-0.253)</i>	0.068 <i>(0.012-0.124)</i>	0.159 <i>(0.096-0.223)</i>
F1	0.557 <i>(0.522-0.588)</i>	0.583 <i>(0.546-0.614)</i>	0.527 <i>(0.501-0.549)</i>	0.578 <i>(0.548-0.611)</i>
ACC	0.563 <i>(0.537-0.593)</i>	0.582 <i>(0.543-0.617)</i>	0.528 <i>(0.504-0.555)</i>	0.574 <i>(0.544-0.603)</i>
<b>Valence</b>				
	ABOY	DBD	ABOY-BNDK	DBD-PKB
MCC	0.102 <i>(0.037-0.166)</i>	0.153 <i>(0.082-0.232)</i>	0.115 <i>(0.053-0.172)</i>	0.149 <i>(0.074-0.238)</i>
F1	0.558 <i>(0.527-0.593)</i>	0.588 <i>(0.552-0.623)</i>	0.565 <i>(0.528-0.598)</i>	0.567 <i>(0.541-0.600)</i>
ACC	0.548 <i>(0.519-0.577)</i>	0.57 <i>(0.538-0.606)</i>	0.552 <i>(0.524-0.579)</i>	0.569 <i>(0.535-0.610)</i>

Table 5.5: Classification performances of four different pipelines for the WCS-Medical subset 10x5 CV scheme.

### 5.2.2 Performances on WCS-Medical

As a second step, accuracy for the WCS-Medical was computed with different pipelines. Comparable performance was found between the new dataset and DEAP (see Table 5.5). However, state-of-art algorithms (i.e. ABOY and BEN) perform slightly worse on WCS-Medical dataset.

### 5.2.3 Performances on WCS-WDs

Finally, we investigate classification performance between the medical-grade and WD subsets (see Table 5.6 for overall performances).

For Arousal, the performances of the ABOY and DBD algorithms on the WCS-E4 subset are comparable, and similar to state-of-art results. For Valence, on the WCS-E4 subset the ABOY algorithm provides slightly better performances than DBD, differently from Arousal. In both cases no

<b>Arousal</b>			
	(E4) ABOY	(E4) DBD	(HB) ADPK
MCC	0.118 <i>(0.071-0.178)</i>	0.129 <i>(0.062-0.214)</i>	0.193 <i>(0.138-0.248)</i>
F1	0.555 <i>(0.527-0.589)</i>	0.563 <i>(0.535-0.598)</i>	0.596 <i>(0.568-0.623)</i>
ACC	0.554 <i>(0.531-0.584)</i>	0.559 <i>(0.527-0.598)</i>	0.59 <i>(0.563-0.616)</i>
<b>Valence</b>			
	(E4) ABOY	(E4) DBD	(HB) ADPK
MCC	0.158 <i>(0.086-0.248)</i>	0.127 <i>(0.046-0.220)</i>	0.136 <i>(0.079-0.198)</i>
F1	0.594 <i>(0.566-0.623)</i>	0.562 <i>(0.523-0.607)</i>	0.561 <i>(0.530-0.590)</i>
ACC	0.57 <i>(0.538-0.610)</i>	0.559 <i>(0.521-0.603)</i>	0.561 <i>(0.535-0.589)</i>

Table 5.6: Classification performances achieved on the WCS-E4 and WCS-HB subsets with 10x5 CV scheme according to different pipelines.

significant differences have been found between the two datasets (WCS-Medical and WCS-E4) or between the two pipelines (DBD and ABOY).

Regarding the WCS-HB subset, similar results have been found: performances for both Arousal and Valence are not statistically different from those found on the WCS-Medical subset; notably, the Comftech HeartBand device achieved the best performance in the detection of Arousal.

### 5.3 Discussion

The validation study presented shows that the results obtained from WDs and medical-grade devices are comparable. However, it is worth highlighting that high performance WDs have been used, thus this result might not hold for lower level devices.

Moreover, participants were required to stay still during the experiment, in order to avoid the artifacts caused by body movements, and provide golden-standard quality for signals acquired by WDs. Before adopting WDs for applications in real-life scenarios, validation of WDs should also consider effects of body movements. This aspect has been partially addressed in the validation of the DBD algorithm (see Appendix A), but additional investigation is required.

With the only exception of the detection of Valence with the Empatica E4, in the general case, the application of our novel algorithms provides better results. However, to determine whether improvements are significant or not, a statistical analysis should be performed on a larger sample size.

These findings do not only assess the possibility of using WDs to reproduce affective computing results, they also provide a proof-of-concept for real-life emotion recognition applications, for instance in psychological therapy. However, additional evaluations are required to ensure an adequate performance of the whole signal acquisition and processing chain in a real-life use case, for instance in terms of latency between the generation of the data and the feedback, computation load and processing time. The outcome is expected to provide key indications also to drive the implementation of the system, for instance whether to move part or the whole processing phase from the server to the local device (e.g. tablet) or sensor.

Despite that accuracy was found comparable with published results on DEAP, the overall performance in terms of automatic emotion recognition is not yet satisfying for real use applications. Music-video stimuli might well not elicit a strong emotional state and, consequently, a clear physiological response; in addition, variability in the musical tastes of participants and mood might also contribute to the overall performance. Therefore, further research is needed before such emotion recognition task could be applied to real-life use cases, considering also that the noise of the signals

is expected to increase when the subject is allowed to move freely.

However, the signal processing algorithms adopted to extract the physiological features deserve a separate consideration. Their validity and reliability have been evaluated separately (see Appendices A and B) proving that they can be applied to investigate the psycho-physiological response in the diverse real-life experimental scenarios, as described in the four use cases presented in Chapter 6.

In summary, due to the difficulty of the task, the WCS dataset can be an appropriate benchmark dataset to demonstrate the efficiency of new algorithms. Different machine learning models can be tested; in particular, deep learning methods are expected to improve the overall results, as already demonstrated in different fields.

It is worth noting that we fully exploited data analytics techniques (e.g. signal processing, machine learning) and technological resources (e.g. PhysioREC, Physiolitix, pyPhysio) in the developed framework to validate the usage of WDs in research.

In Chapter 6 we indeed present two research studies that motivated the development of such framework, which are focused differently on the methodological and technological aspects. Further, we describe other two recent studies that are based on the results on the novel data resource presented in this Chapter and employ WDs to investigate physiological response: in parenting and music-therapy.



# Chapter 6

## Case studies and applications

In this Chapter we discuss four studies which highlight the interest of Wearable Devices (WDs) in research. The first study (Section 6.1) is focused on HRV analysis, and addresses the need of a customizable signal processing tool which could be integrated in more complex pipelines. The results are based on the pyHRV Python package, which is a key solution available with pyPhysio, and thus in Physiolyze, the web-interface for signal processing used as a main building block for the Physiolitix project.

Physiolitix has been applied for the first time for outdoor acquisition to detect fatigued states in skiers, presented in Section 6.2. The third and fourth studies discussed in this Chapter make use of WDs to collect physiological signals in a therapeutic context, where the usage of medical-grade devices is not feasible. Specifically, the third study (Section 6.3) aims at evaluating the physiological response of parents to different types of crying of children affected by Autism Spectrum Disorders (ASD). The fourth study (Section 6.4) is an ongoing project that aims to identify interpersonal synchrony between a therapist and a child with ASD during a music-therapy session.

## 6.1 Heart Rate Variability pipeline to investigate response to maternal care

Previous studies (e.g. [40]) showed that the psycho-physiological state of the newborn adapts to maternal care, in both humans and mice, which has a calming effect in terms of crying episodes and body movements. Moreover, an effect on autonomic regulation was also hypothesized as the researchers observed a significant difference in the average IBIs.

To investigate this aspect, we propose a tool to perform Heart Rate Variability analysis. In this section we illustrate and validate pyHRV and Physiolyze [17] to obtain an Heart Rate Variability analysis integrated into a machine learning pipeline. These results have been instrumentals in the development of pyPhysio and Physiolytix (see Chapters 2 and 3).

This work has been conducted in collaboration with the RIKEN Brain Science Institute (Tokyo, Japan), and presented at the 8<sup>th</sup> Conference of the European Study Group on Cardiovascular Oscillations. An improved version of Physiolyze with additional real-time processing features has been presented at the 2014 Galaxy Community Conference.

### 6.1.1 Aim of the study

The goal of the study is to explore the possibility to integrate HRV analysis into a machine learning (ML) pipeline. Existing tools for HRV analysis can be grouped in open-source libraries (PhysioToolkit [49], RHRV [101]) and offline platforms (KubiosHRV [120], gHRV [100]). None of the existing solutions allowed the integration in more complex pipelines while featuring ease of use also for non-experts.

To this aim, we first designed a tool that integrates HRV analysis into a comprehensive ML pipeline without requiring advanced data science and programming skills. The pipeline is then validated on a dataset of IBI series

of infant data to investigate autonomic regulation in different behavioral states.

### 6.1.2 Materials and methods

ECG signals from 25 infants were collected with a Light WP Holter ECG recorder (GE Healthcare) as described in [40]. IBIs were automatically computed by this device. Three behavioral states were annotated: sleep (SL), awake and calm (CM), awake and crying (CY). The analysis here is restricted to a complete set of data from 7 subjects, for which all behavioral states are available in the same experimental session.

To allow user-friendly HRV analysis and predictive modeling we developed Physiolyze, which is composed by two modules:

1. the Python package pyHRV, which implements the core algorithms used for the HRV analysis;
2. a Galaxy-based [48] user interface to allow pipeline customization and execution.

The Python library pyHRV provides algorithms for extracting about 40 up-to-date HRV indexes built on top of aforementioned tools further evolved to cover non-linear and entropy-based indexes [3, 90].

Physiolyze fully exploits Galaxy [48] features, in particular its workflow editor, and the support for reproducible ML experiments (see Figure 6.1), adding specialized functions for extraction of HRV indicators provided by pyHRV.

IBI signals are segmented by windowing (length of window: 40 s, shift between consecutive windows: 10 s) and 24 indicators (time and frequency domain and non-linear metrics) were extracted for each segment. Based on original annotations of infant behavioral state, each window is then assigned to one of the target classes, namely S, CM and CY. Windows

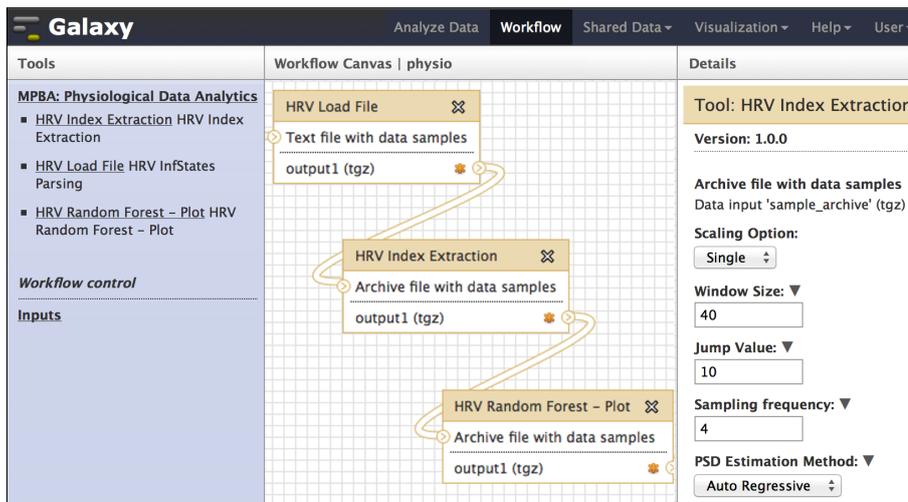


Figure 6.1: An HRV analysis pipeline available through the Physiolyze web interface. Three processing steps (preprocessing, HRV index extraction, predictive modeling) are linked together to form the workflow in the central panel. Panel on the right allows setting module parameters: parameters for HRV index extraction are shown here.

overlapping different states are rejected. As a result of the HRV Index Extraction step (see Figure 6.1), a dataset has been produced in which each sample derives from a segment of the original IBI signal, the features are the 24 HRV indicators and the label is the class to which the segment belongs.

This dataset has been used as input to the following machine learning step. In this specific case, the selected predictive model is a Random Forest [19], implemented with the *randomForest* R library, as it allows to deal with multi-class problems. The goal was to predict infant behavioral state from the HRV indicators. This study also prompted the development of a working example of integration of tools for physiological signal processing and machine learning with a user-friendly platform.

### 6.1.3 Results

Results from classification of HRV states show good applicability of pyHRV for extracting state-of-art HRV indicators. A total of 1178 samples are

Label	#samples	Predicted			Class Error [%]
		SP	CM	CY	
SP	367	<b>295</b>	72	0	19.6
CM	604	48	<b>531</b>	25	12.1
CY	207	2	50	<b>155</b>	25.1

Table 6.1: Confusion matrix from internal Out-Of-Bag of RandomForest classification of infant behavioural states.

extracted from IBI signals of 7 infants and merged in the same dataset after subject-wise normalization. Classes are not balanced; most samples are from the CM class (604 samples) which represents more than half of the dataset. Remaining samples are divided into SP class (367 samples) and CY class (207). Based on the HRV indicators we have been able to predict behavioral states with 83.3% of accuracy (see Table 6.1) demonstrating potential of HRV analysis to highlight psycho-physiological variations in infants.

#### 6.1.4 Conclusions

In this study we introduce pyHRV and Physiolyze, two resources we developed to enable the integration of HRV analysis into articulated processing pipelines. The development paradigm we adopted fosters the use of open-source and user-friendly tools to allow researchers with limited programming skills to develop their own pipelines. The whole framework has been successfully tested on a classification task aiming at predicting the infant behavioral states.

In particular, this study ignited the development of pyPhysio and Physiolitix as complementary components of the same framework: the first featuring algorithms for analysis of multi-modal datasets, the second as a technological platform to allow application of these capabilities to experimental use cases in real world scenarios.

## 6.2 Detection of fatigue in skiers from inertial signals

Skiing is a potentially risky sport and the safety of skiers is a very important matter. As a general prevention strategy, several studies proposed to stop skiing before fatigue could negatively affect the physical capabilities and increase the risk of injuries [56, 66], however the role of fatigue in ski accidents has been only partially investigated. In particular, the relationship between fatigue and injuries has been analyzed only for specific injuries (e.g. Anterior Cruciate Ligament injuries [105]), but it remains in general unclear the critical level of fatigue that could be used as indicator to prevent accidents.

Thanks to an existing project based on the SicurSkiWeb platform<sup>1</sup>, close to 15000 events involving injuries have been collected to investigate the temporal and spatial distribution of ski accidents on Trentino ski areas. The study showed that ski accidents are more likely to occur in specific times of the day and locations in the ski-areas. In particular, late morning hours (pre-lunch and lunch hours) showed the higher number of accidents.

In this section, we present a preliminary study aiming at investigating effects of fatigue on movement patterns, which was presented in the 21<sup>st</sup> International Conference of the International Society for Skiing Safety (2015).

### 6.2.1 Aim of the study

In downhill skiing, fatigue is expected to affect muscular reactivity and control, thus altering rapid response trajectories. The hypothesis is that an high number of accidents is caused by fatigue states that skiers experience after some hours of skiing, which affects physical and cognitive performances and responsiveness.

---

<sup>1</sup><https://sicurskiweb.fbk.eu/>

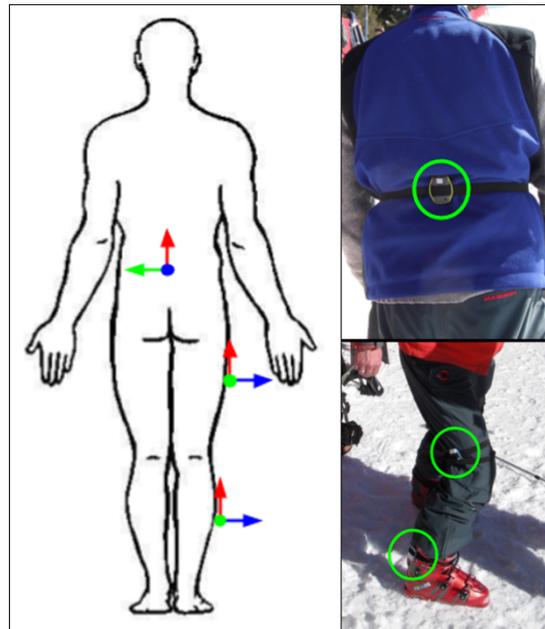


Figure 6.2: Setup of the Body Sensor Network composed of 3 Inertial Measurement Units: on the back, right thigh and right leg. Left: orientation of sensors; right: positioning during a pilot test.

In this study we aimed at exploring the usage of WDs to provide early indicators of fatigue, in particular by defining a predictive model of fatigue based on inertial data and position on-the-field.

### 6.2.2 Materials and methods

Collection of inertial data was based on a body sensor network composed by three Inertial Measurement Units (IMUs) EXEL Exls3 and an Android smartphone as Body Central Unit.

Each IMU embeds a triaxial accelerometer with selectable full-scale range set to  $\pm 16g$ , a triaxial gyroscope (range set to:  $\pm 2000dps$ ) and a triaxial magnetometer. Sampling frequency was set to 100 Hz. Signals are sent via Bluetooth 2.0 to the BCU where the PhysioREC app verifies missing-data events, synchronizes, and stores data flows on the smartphone. Position data from smartphone embedded GPS sensor (sampling

frequency 1 Hz) were also collected. IMUs were worn by the skier by means of elastic bands positioned at lower back, right thigh and right leg (see Figure 6.2).

Five skiers with varying skiing experience (from 2 to >15 years) were involved in the data acquisition in the Monte Bondone ski area, Trento, Italy. During the first run skiers were free to choose an easy or intermediate slope to warm up. Inertial data were collected during second and last run which was mandatorily on the same track for all subjects. Otherwise, skiers were free to ski in the whole ski area, with GPS tracks automatically annotated. The duration of total experimental session was about 180 minutes. As connectivity to the internet was not available on the whole ski area, data were stored on the smartphone and on the Exls3 internal memory as a backup.

We focused on the data collected by the IMU positioned on the thigh, as it resulted as the most stable during all the acquisition sessions. Collected data were preprocessed with a custom pipeline based on pyPhysio (a summary can be found in Table 6.2):

1. Selection of ski segments: we distinguished data associated to ski from other activities (ski-lift, resting). Skiers were asked to hit the ground with the foot three times immediately before and after the run. Hits are well recognizable from the acceleration signal and were used to identify ski portions;
2. Low pass filtering, with an Infinite Impulse Response filter, cut-off frequency set at 25 Hz;
3. Computation of the module for each sensor as additional signal;
4. Derivative based on 1<sup>st</sup>, 2<sup>nd</sup>, 5<sup>th</sup> and 10<sup>th</sup> order differences;
5. Segmentation by windowing: length of the window: 15 s, shift between

Name	Type	Number
IMU	Thigh	1
Sensors	Accelerometer, Gyroscope, Magnetometer	3
Signals	3 axis and module	4
Processed signals	Original, 1-2-5-10 <sup>th</sup> order differences	5
Inertial features	Time domain and Frequency domain	28

Table 6.2: Summary of input data for the detection of fatigued states from inertial signals.

Name	NF	F	Total
Subject 01	14	11	25
Subject 02	20	24	44
Subject 03	43	35	78
Subject 04	16	14	30
Subject 05	22	19	41

Table 6.3: Number of samples for each subject and class.

consecutive windows: 5 s. Each segment was labeled as Non-Fatigued state (NF) or Fatigued state (F) depending on whether the signal was acquired during second or last run;

6. Computation of inertial features (8 time-domain and 20 frequency domain) for each preprocessed signal for a total of 1680 features for each segment.

A total of 218 segments were obtained: 115 belonged to NF class, 103 to F class; the number of samples for each subjects is shown in Table 6.3.

The objective of the analysis was to develop a predictive model able to classify the state of each segment based on the 1680 features.

The Machine Learning (ML) pipeline was based on Support Vector Machines (SVM) models and composed of three nested randomization cycles (see Figure 5.1): first cycle corresponds to the Cross Validation (CV) scheme, second to the tuning of model parameter C, finally the feature selection cycle. Features were ranked internally based on SVM weights. We

used the Matthew Correlation Coefficient (MCC) as metrics of classification performance, 95% confidence intervals (CI) were computed by Student t-test bootstrapping of MCC performances from each cross-validation cycle. Maximal average MCC of all feature selection steps is reported as performance index of each model.

### 6.2.3 Results

In the first experiment, samples from all subjects were merged in the same dataset and a 10x5 CV scheme was adopted. Maximal average MCC of 0.895 (95% CI: 0.878 - 0.911) was reached with 900 features, however 90 features were enough to obtain an average MCC of 0.823 (95% CI: 0.806 - 0.840).

In the second experiment we tested whether the predictive model can be transferred and applied to new subjects, using a Leave-One-Subject-Out test scheme (LOSO experiment). To simulate the application of the model to new data (e.g. a new subject participating in the study or a new ski session), all samples belonging to a subject were removed from the input dataset. The remaining samples (from 4 subjects) were used to build a new predictive model applying the same pipeline as in the first experiment. The resulting model was used to predict fatigued state on the samples of the subject left out. The procedure was then repeated for each subject.

We observed a drop in prediction performances of left-out subjects (see Table 6.4), showing that the model can be easily biased, possibly by overfitting the input dataset and thus losing generalizability.

Therefore, we computed the intersection of the top 500 features for each LOSO experiment, obtaining a list of 117 features which were used to create a reduced dataset. New LOSO experiment based on the reduced dataset dramatically improved the classification performances (see Table 6.4 demonstrating that it would be feasible to apply the predictive model

Left-out subject	MCC	
	Complete dataset	Reduced dataset
Subject 01	-0.103	0.541
Subject 02	0.431	0.797
Subject 03	0.182	0.617
Subject 04	0.092	0.782
Subject 05	0.488	0.910
<i>Average</i>	<i>0.218</i>	<i>0.730</i>

Table 6.4: Summary of input data for the detection of fatigued states from inertial signals.

also to new subjects. To check for a possible selection bias in the model design, the same experiment was repeated using randomly shuffled sample labels, obtaining average MCC values around 0, thus proving that the obtained models were not biased.

#### 6.2.4 Conclusions

Results of this preliminary study demonstrate feasibility of identification of early indicators of fatigue on skiers by detecting changes in movement patterns. Predictive models appeared more efficient when specifically tuned on each subjects, however we also tested generalizability showing that by accurately selecting input features it is possible to reach good classification performance even on data from new subjects. This work could then represent a valid methodological approach for preventing accidents in real-life applications, but before adoption on large-scale trials new studies should be performed with a larger sample size, possibly accounting for diverse ages and skill levels and adopting an objective measure of fatigue. In fact, it is likely that fatigue affects each subject differently, according also to conditioning, experience and other physical characteristics. Our general assumption is that the fatigue level during the last run was greater than in the second run; however, according to [121, 127], an alternative measure of fatigue could be the blood lactate concentration, which is expected to

be a more reliable physiological indicator.

In addition, the same approach should be tested on inertial data from the smartphone which would allow the adoption in real-world applications. Smartphones would represent an optimal solution as they provide both sensing and computational capabilities which would allow real-time processing and a simplified sensing architecture, with any need of external IMUs. However, it should be first verified whether lower signal quality prevents the appropriate accuracy in the detection of the different motor patterns. For a real-world application based on a smartphone, both as sensing and as computational resource, the number of computed features might be decreased, for instance focusing on the top 117 features resulting from the LOSO experiment. This would potentially reduce the computational load and battery consumption.

In fact the computation and time limitations can strongly influence the applicability of the system to real use cases, but in this work these aspects have not been considered due to the preliminary nature of the study. As additional solution, part of the preprocessing can be moved from the smartphone to the IMU device in case this is equipped with an onboard architecture with a Computational Processing Unit. For instance, as proposed in [22] a context-aware module can be implemented to optimize the battery consumption by enabling the sensor only during skiing.

Finally, a complete investigation should now consider the adoption of deep learning models which have been demonstrated to outperform state-of-art results obtained from standard ML models (e.g [92, 82]. In particular, subject-based models or transfer learning techniques might be able to model better the individual characteristics and thus further improve the results.

Beside results on fatigue detection, this study also validated the use of PhysioREC for on-the-field acquisition of multi-sensor data. In particular,

with PhysioREC it was possible to simultaneously activate all sensors and synchronize the signals, managing also experimental metadata (such as: ID of the subject, ID of the run) and integrative position information. Real-time upload on the Physiokitix back-end was not implemented in the version used for this study, due to lack of data connection on the entire experimental area.

It is worth mentioning that the experimental contexts were dramatically different from standard in-lab settings: acquisition sessions were performed during night-time public opening hours, without use of any supporting aid (e.g.: mobile lab, deposit, heated rooms). In addition, setting and control of the experiment was up to the skiers who were instructed after a short briefing before each session. This demonstrate reliability of PhysioREC as a flexible tool for real-life sensing even in critical outdoor contexts.

### 6.3 Parental response to infant crying

Crying is one of first means of communication in humans [39]. During early development of infants, parents learn how to interpret crying to intervene and assist the child [72]. However, the correct interpretation of crying might be prevented, because of inadequate capacity of the parent or because of atypical acoustic patterns in crying. When this happens it could originate issues in the child-parent relationship [38].

This is frequent in case of children affected by Autism Spectrum Disorders (ASD): atypical patterns of distress vocalization are evident even before ASD are diagnosed [37]. Crying of children affected by ASD (ASD crying) elicits more negative emotions which might prevent the correct parent response and feedback.

Several studies focus on perceived emotions of parents hearing typical and atypical (e.g. ASD) crying but few is known about the physiological

and autonomic response, which is the object of investigation of this study.

This work<sup>2</sup> has been submitted to the Journal of Autism Spectrum Disorders.

### 6.3.1 Aim of the study

In this study we focused on the physiological response of parents to crying, considering two dimensions:

1. Type of parents: parents of children affected by ASD (pASD) and parents of typically developing children (pTD);
2. Type of crying: crying of children affected by ASD (ASDc) and of typically developing children (TDc).

In particular, we investigated: (a) whether ASDc and TDc elicit different physiological responses on parents, and (b) whether pASD and pTD perception of two types of crying differs.

### 6.3.2 Materials and methods

Participants were grouped in pTD (30 participants: 15 mothers and 15 fathers) and pASD (19 participants: 11 mothers and 8 fathers). Crying stimuli (duration 6 seconds) were extracted from home videos of infants of similar age to compose a set of 8 ASDc and 7 TDc were obtained. Furthermore, two questionnaires (the Parenting Stress Index-Short Form and the Weinstein Noise Sensitivity Scale) were administered to examine possible correlations between parents responses to crying and their parenting stress or noise sensitivity. No correlation was found.

---

<sup>2</sup>YAGMUR OZTURK, ANDREA BIZZEGO, GIANLUCA ESPOSITO, CESARE FURLANELLO, AND PAOLA VENUTI, Response to infant cry: Self-report and physiological measures from parents of children with Autism Spectrum Disorder. *Under review*

The experiment comprised two parts: the first was designed to acquire physiological signals, the second to collect behavioral data.

In the first part, 32 crying stimuli (16 ASDc and 16 TDc) were randomly sampled from the set of 8 ASDc and 7 TDc and randomly ordered. Each 6 seconds stimulus was followed by 10 seconds of pause. Participants were asked to stay still during the experiment to avoid moving artifacts. Empatica E4 was used as the device to acquire physiological signals. We chose the WD as we needed flexibility in terms of experiment setup, since parents of children affected by ASD could participate to the experiment only during the therapeutic sessions of their own child and the structure was not equipped with medical grade devices.

Inter Beat Intervals (IBIs) were extracted from the Blood Volume Pulse (BVP) signal acquired from the Empatica E4 through the DBD algorithm (see Appendix A). Average IBI (IBImean) was then computed on each segment corresponding to a crying stimulus. We also computed the baseline IBImean during the 30 seconds portion before the start of the first stimulus, which was used to normalize stimuli IBImean values.

During the second part of the experiment each stimulus was presented once and the participant was asked to rate the perceived emotion according to three dimensions: Stress, Arousal and Valence.

### 6.3.3 Results

No significant differences were found in self-reported emotion between the type of parents when listening to ASDc or TDc. Instead, considering the same parent category significant differences have been found between the type of crying. In particular, for both categories of parents, ASDc has been reported as more stressful, more arousing and less pleasant than TDc.

Opposite to results from self-reports, significant ( $p < 0.05$ ) differences in IBImean were found within the same type of crying: for both categories

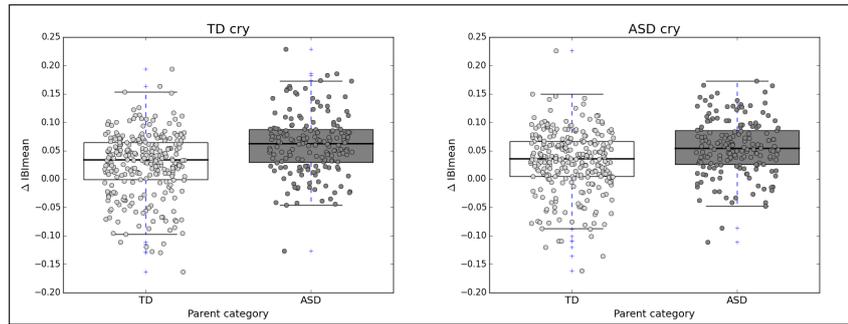


Figure 6.3: Differences in IBImean between types of parents.

(ASDc and TDc), pASD showed higher IBImean than pTD (see Figure 6.3).

### 6.3.4 Conclusions

Although no difference was found in self-reported emotion between types of parents, analysis of physiological indicators evidenced a different response in terms of heart rate. This result demonstrates the importance of considering both the emotional and physiological aspects as they can give complementary information. In particular, future studies should clarify whether physiological variations are also associated to differences in terms of autonomic regulation.

Beside shading new light on the understanding of parent response to infant crying, this study also proved that WDs can be reliably used as a flexible tool to collect physiological signals when contextual constraints limit the use of medical-grade devices. In addition, it validates the use of pyPhysio as signal processing tool for the extraction of physiological indicators.

We also demonstrated potential of complementary physiological monitoring to expand comprehension of human behavior. In combination with WDs, this approach paves the way for new applications in treatment of psychological pathologies and training of therapists which is the objective

of the research project presented in next section.

## 6.4 Synchrony in music-therapy of children with ASD

Music-therapy has been widely adopted to help children affected by Autism Spectrum Disorders (ASD) to improve social interaction and communication skills, and emotional reciprocity [46]. Music-therapy aims at building and maintaining a relationship between the therapist, however the administration of the treatment and choice of techniques are still based on the subjective evaluation of the therapist. Few studies aimed at investigating music-therapy to provide quantitative indicators to identify most appropriate techniques to improve efficiency of the treatment [115, 116].

In this research project, conducted in collaboration with ODFLab of University of Trento, we aim at quantifying the level of interaction between the therapist and the child following a recent approach based on physiological synchrony [81].

### 6.4.1 Materials and methods

Electrocardiogram (ECG) signal is acquired during therapy sessions from both the therapist and the child using the ComfTech HeartBand device within a multi-subject body sensor network managed by PhysioREC and Physiolitix. During the therapy, the therapist communicates (by mean of an appropriate body sign) to an external observer the instants when an high level of interaction is perceived, which are annotated using the marker button on the PhysioREC user interface. Adoption of WDs in treatment of ASD has been already proposed in other studies [34, 21] to support patient deficits in recognizing and expressing emotions, by identification of autonomic states through physiological indicators, though recognizing

synchrony during music-therapy corresponds to approaching this aspect from a social perspective and it is complicated by the dynamic context of the therapy.

From the ECG signal of each subject we will compute the IBIs, then HRV indicators will be computed on subsequent portions of the IBI signal extracted by windowing (window length: 60 s, shift between consecutive windows: 15 s). Physiological synchrony will be quantified estimating the similarity between IBI signals from the therapist and the child along the therapy session. HRV indicators will also be considered, though accounting for lower reliability of frequency domain indicators. To this aim, the similarity between the two IBI signals will first be computed for each segment extracted by windowing (window length: 60 s, shift between consecutive windows: 15 s). Diverse metrics will be adopted to assess similarity between time series, for instance: Maximal Information Criterion [98], Dynamic Time Warping (DTW) [29], Recurrence Plot analysis [81].

### 6.4.2 Initial and expected results

The hypothesis of this research is that social and behavioral interaction between the therapist and the child is associated to synchrony of the physiological patterns mediated by the autonomic response. Thus, we expect to find higher similarity between IBI signals in correspondence of annotated instants of perceived high level of interaction. To validate findings we will statistically compare similarity metrics during baseline (for instance, start of the therapy) and during instants of perceived interaction.

A pilot study was conducted with two therapy sessions, involving a therapist and two children simultaneously. Physiological synchrony was computed by mean of DTW distance for each dyad: Therapist-Child1, Therapist-Child2, Child1-Child2 (see Figure 6.4). A preliminary analysis of the therapy and resulting synchrony metrics showed that specific activities



Figure 6.4: A frame of the video of the music-therapy session, showing two children with ASD (left and right) and the therapist (middle) while playing drums. Graphs on the left-bottom of the figure show the average IBI (RR mean) for each subject: therapist (blue), child on the right (green) and child on the left (red) during the entire session. The instant corresponding to the showed frame is identified by the vertical red line. Vertical bars on the bottom-right show current levels of synchrony: between Therapist and Child1 (left), between Therapist and Child2 (middle) and between the two children (right). Here an high level of physiological synchrony is found between the two children who are coordinated in playing drums together.

(e.g. playing drums) which facilitate the behavioral synchrony appear also associated to a higher level of physiological synchrony.

Our results from this study should provide a first insight on the physiological contribution to therapeutic outcome and provide indications to develop a biofeedback system that notifies the therapist when the level of interaction is optimal. Applications are not limited to music-therapy as monitoring the level of interaction would also benefit child carers, parents first but also teachers and relatives.



# Chapter 7

## Conclusions

WDs represent an opportunity for research, as they pave the way for acquisition of physiological signals in new experimental contexts. Moreover, the application of WDs would allow for real-life continuous monitoring, and enable acquisition of physiological signals even when usually the adoption of a complete laboratory settings is unfeasible. Beside the several advantages introduced by the technological progress (such as miniaturization, portability, increasing autonomy), there are indeed limitations and constraints imposed by low quality sensors, low signal noise ratio and sensitivity to artifacts due to body movements. Furthermore, the variability in protocols to access the data and to control the different devices prevents the integration of heterogeneous WDs into the same sensing architecture. These aspects prevented the adoption of WDs in research, despite the growing and growing interests of commercial applications.

In this thesis, we concentrated on two main aspects: feasibility of physiological signal acquisition for research based on WDs and assessment of capability of reproducing state-of-art results.

The first result discussed is a platform for real-life sensing named Physiolitix, which is composed by PhysioREC, an Android app to manage Body Sensor Networks, and a back-end which centralizes and provide access to

the data collected from sensors. Thanks to its modular structure, PhysioREC is potentially capable of collecting data from all types of WDs for which manufacturers provide the data streaming protocol documentation or a corresponding SDK. Moreover, as streams from WDs are simultaneously collected, PhysioREC is capable of synchronizing the signals and control the whole Body Sensor Network. Complementary to PhysioREC, Physiolitix back-end is where streams are stored, useful for researchers to retrieve from a single place all experimental data.

Thanks to Physiolitix we were able to create the Wearable and Clinical Signal (WCS) dataset to validate the WDs. The WCS dataset reproduces the experimental setup adopted to create the DEAP dataset, using WDs as additional sources of physiological signals, thus allowing a direct comparison of results from this study with the scientific background. The WCS dataset is the first example of dataset where physiological signals from wearable and medical-grade devices are collected simultaneously. The WCS dataset can be adopted by manufacturers and researchers as a benchmark to validate new devices or algorithms.

Based on the WCS dataset, we finally investigated whether WDs are able to reproduce scientific results achieved with medical-grade devices. The validation focuses on an affective computing task in which the emotional content of music-video stimuli are recognized from physiological indicators. The study involves the comparison of classification performances of different predictive models.

Results suggest that it is possible to replicate state-of-art results using signals from WDs by applying proper signal processing methods. In particular, we focused on BVP and EDA signals and on the case in which artifacts from body movements are experimentally avoided. We also report that the proposed algorithms for signal processing can improve the classification and that the WCS dataset reproduces the original DEAP

dataset. The signal processing was based on pyPhysio, developed to allow the analysis of heterogeneous types of physiological signals within a unique framework.

Several research fields might benefit from the outcome of this work. We present an application in an outdoor context, in which Physiolitix was used to collect inertial data from skiers and predict fatigued states. We also describe an application of WDs adopted in less extreme contexts, to substitute medical-grade devices when they are not available or cannot be used. In this context, we used a WD to investigate physiological response of parents to infant crying in terms of Heart Rate Variability. An ongoing research project aims at identifying physiological indicators of synchrony between a therapist and a patient during a music-therapy session.

WDs are expected to become a pervasive technology in the future. This work provides both a technological framework and a methodological contribution to allow exploiting the potential of WDs for reproducible science.



## Bibliography

- [1] M. Aboy, J. McNames, T. Thong, D. Tsunami, M. S. Ellenby, and B. Goldstein. An automatic beat detection algorithm for pressure signals. *IEEE Transactions on Biomedical Engineering*, 52(10):1662–1670, 2005.
- [2] H. Alemdar and C. Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710, 2010.
- [3] F. Aletti, M. Ferrario, T. B. A. de Jesus, R. Stirbulov, A. B. Silva, S. Cerutti, and L. M. Sampaio. Heart rate variability in children with cyanotic and acyanotic congenital heart disease: analysis by spectral and non linear indices. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4189–4192. IEEE, 2012.
- [4] D. M. Alexander, C. Trengove, P. Johnston, T. Cooper, J. August, and E. Gordon. Separating individual skin conductance responses in a short interstimulus-interval paradigm. *Journal of neuroscience methods*, 146(1):116–123, 2005.
- [5] D. R. Bach. A head-to-head comparison of scralyze and ledalab, two model-based methods for skin conductance analysis. *Biological psychology*, 103:63–68, 2014.
- [6] D. R. Bach, G. Flandin, K. J. Friston, and R. J. Dolan. Time-series analysis for rapid event-related skin conductance responses. *Journal of neuroscience methods*, 184(2):224–234, 2009.
- [7] E. Balanou, M. van Gils, and T. Vanhala. State-of-the-art of wearable EEG for personalized health applications. In *PHealth 2013: Proceedings of the 10th International Conference on Wearable Micro and*

*Nano Technologies for Personalized Health*, volume 189, page 119. IOS Press, 2013.

- [8] A. J. Bandothkar, D. Molinnus, O. Mirza, T. Guinovart, J. R. Windmiller, G. Valdés-Ramírez, F. J. Andrade, M. J. Schöning, and J. Wang. Epidermal tattoo potentiometric sodium sensors with wireless signal transduction for continuous non-invasive sweat monitoring. *Biosensors and bioelectronics*, 54:603–609, 2014.
- [9] F. S. Bao, X. Liu, and C. Zhang. Pyeeg: an open source python module for EEG/MEG feature extraction. *Computational intelligence and neuroscience*, 2011, 2011.
- [10] A. Bashan, R. P. Bartsch, J. W. Kantelhardt, S. Havlin, and P. C. Ivanov. Network physiology reveals relations between network topology and physiological function. *Nature communications*, 3:702, 2012.
- [11] D. R. Bassett. Device-based monitoring in physical activity and public health research. *Physiological measurement*, 33(11):1769, 2012.
- [12] A. Battisti. Sensorflow: an integrative approach to sensor data streams. Bachelor’s thesis, University of Trento, Italy, July 2016.
- [13] M. Benedek and C. Kaernbach. A continuous measure of phasic electrodermal activity. *Journal of neuroscience methods*, 190(1):80–91, 2010.
- [14] M. Benedek and C. Kaernbach. Decomposition of skin conductance data by means of nonnegative deconvolution. *Psychophysiology*, 47(4):647–658, 2010.
- [15] J. E. Berlin, K. L. Storti, and J. S. Brach. Using activity monitors to measure physical activity in free-living conditions. *Physical Therapy*, 86(8):1137–1145, 2006.

- [16] A. Bideaux, B. Zimmermann, S. Hey, and W. Stork. Synchronization in wireless biomedical-sensor networks with bluetooth low energy. *Current Directions in Biomedical Engineering*, 1(1):73–76, 2015.
- [17] A. Bizzego, M. Mina, C. Zarbo, G. Esposito, and C. Furlanello. Physiolyze: a Galaxy-based web service for heart rate variability analysis with online processing. In *2014 8th Conference of the European Study Group on Cardiovascular Oscillations*, pages 97–98. IEEE, 2014.
- [18] A. Botta, W. De Donato, V. Persico, and A. Pescapé. On the integration of cloud computing and internet of things. In *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*, pages 23–30. IEEE, 2014.
- [19] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [20] M. M. Bundeale and R. Banerjee. An SVM classifier for fatigue-detection using skin conductance for use in the BITS-lifeguard wearable computing system. In *2009 Second International Conference on Emerging Trends in Engineering & Technology*, pages 934–939. IEEE, 2009.
- [21] J.-J. Cabibihan, H. Javed, M. Aldosari, T. W. Frazier, and H. Elbashir. Sensing technologies for autism spectrum disorder screening and intervention. *Sensors*, 17(1):46, 2016.
- [22] F. Casamassima, E. Farella, and L. Benini. Context aware power management for motion-sensing body area network nodes. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 170. European Design and Automation Association, 2014.
- [23] F. Casamassima, A. Ferrari, B. Milosevic, P. Ginis, E. Farella, and L.

- Rocchi. A wearable system for gait training in subjects with parkinson's disease. *Sensors*, 14(4):6229–6246, 2014.
- [24] G. C. Cawley and N. L. Talbot. Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters. *Journal of Machine Learning Research*, 8(4):841–861, 2007.
- [25] M. Chan, D. Estève, J.-Y. Fourniols, C. Escriba, and E. Campo. Smart wearable systems: Current status and future challenges. *Artificial intelligence in medicine*, 56(3):137–156, 2012.
- [26] B. Chen, J. P. Varkey, D. Pompili, J. K. Li, and I. Marsic. Patient vital signs monitoring using wireless body area networks. In *Proceedings of the 2010 IEEE 36th Annual Northeast Bioengineering Conference*, pages 1–2. IEEE, 2010.
- [27] L. Y. Chen, B. C.-K. Tee, A. L. Chortos, G. Schwartz, V. Tse, D. J. Lipomi, H.-S. P. Wong, M. V. McConnell, and Z. Bao. Continuous wireless pressure monitoring and mapping with ultra-small passive sensors for health monitoring and critical care. *Nature communications*, 5, 2014.
- [28] M. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, and C.-H. Youn. Wearable 2.0: Enabling human-cloud integration in next generation healthcare systems. *IEEE Communications Magazine*, 55(1):54–61, 2017.
- [29] P. Chikersal, M. Tomprou, Y. J. Kim, A. Woolley, and L. Dabbish. Deep structures of collaboration: Physiological correlates of collective intelligence and group satisfaction. In *Proceedings of the 20th ACM Conference on Computer-Supported Cooperative Work and Social Computing*, 2017.

- [30] J. Choi and R. Gutierrez-Osuna. Using heart rate monitors to detect mental stress. In *2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, pages 219–223. IEEE, 2009.
- [31] S. Y. Chung and H. J. Yoon. Affective classification using bayesian classifier and supervised learning. In *2012 12th International Conference on Control, Automation and Systems*, pages 1768–1771. IEEE, 2012.
- [32] M. Consortium and others. The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. *Nature biotechnology*, 28(8):827–838, 2010.
- [33] C. Crema, A. Depari, A. Flammini, and A. Vezzoli. Efficient R-peak detection algorithm for real-time analysis of ECG in portable devices. In *2016 IEEE Sensors Applications Symposium*, pages 1–6. IEEE, 2016.
- [34] S. Di Palma, A. Tonacci, A. Narzisi, C. Domenici, G. Pioggia, F. Muratori, L. Billeci, and MICHELANGELO study group. Monitoring of autonomic response to sociocognitive tasks during treatment in children with autism spectrum disorders by wearable technologies: A feasibility study. *Computers in biology and medicine*, 2016.
- [35] M. Díaz, C. Martín, and B. Rubio. State-of-the-art, challenges, and open issues in the integration of Internet of Things and cloud computing. *Journal of Network and Computer Applications*, 67:99–117, 2016.
- [36] R. J. Ellis, B. Zhu, J. Koenig, J. F. Thayer, and Y. Wang. A careful look at ECG sampling frequency and R-peak interpolation on short-

- term measures of heart rate variability. *Physiological measurement*, 36(9):1827, 2015.
- [37] G. Esposito, J. Nakazawa, P. Venuti, and M. H. Bornstein. Componential deconstruction of infant distress vocalizations via tree-based models: A study of cry in autism spectrum disorder and typical development. *Research in Developmental Disabilities*, 34(9):2717–2724, 2013.
- [38] G. Esposito and P. Venuti. How is crying perceived in children with autistic spectrum disorder. *Research in Autism Spectrum Disorders*, 2(2):371–384, 2008.
- [39] G. Esposito and P. Venuti. Comparative analysis of crying in children with autism, developmental delays, and typical development. *Focus on Autism and Other Developmental Disabilities*, 24(4):240–247, 2009.
- [40] G. Esposito, S. Yoshida, R. Ohnishi, Y. Tsuneoka, M. del Carmen Rostagno, S. Yokota, S. Okabe, K. Kamiya, M. Hoshino, M. Shimizu, and others. Infant calming responses during maternal carrying in humans and mice. *Current Biology*, 23(9):739–745, 2013.
- [41] R. R. Fletcher, M.-Z. Poh, and H. Eydgahi. Wearable sensors: opportunities and challenges for low-cost health care. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pages 1763–1766. IEEE, 2010.
- [42] J. Y. A. Foo. Comparison of wavelet transformation and adaptive filtering in restoring artefact-induced time-related measurement. *Biomedical signal processing and control*, 1(1):93–98, 2006.

- [43] C. A. Frantzidis, C. Bratsas, C. L. Papadelis, E. Konstantinidis, C. Pappas, and P. D. Bamidis. Toward emotion aware computing: an integrated approach using multichannel neurophysiological recordings and affective visual stimuli. *IEEE Transactions on Information Technology in Biomedicine*, 14(3):589–597, 2010.
- [44] M. Garbarino, M. Lai, D. Bender, R. W. Picard, and S. Tognetti. Empatica E3 - A wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition. In *2014 EAI 4th International Conference on Wireless Mobile Communication and Healthcare*, pages 39–42, November 2014.
- [45] G. Garcia-Molina, T. Tsoneva, and A. Nijholt. Emotional brain-computer interfaces. *International journal of autonomous and adaptive communications systems*, 6(1):9–25, 2013.
- [46] M. Geretsegger, C. Elefant, K. A. Mössler, and C. Gold. Music therapy for people with autism spectrum disorder. *The Cochrane Library*, 2014.
- [47] P. Gibbs and H. H. Asada. Reducing motion artifact in wearable biosensors using MEMS accelerometers for active noise cancellation. In *Proceedings of the 2005 American Control Conference*, pages 1581–1586. IEEE, 2005.
- [48] J. Goecks, A. Nekrutenko, and J. Taylor. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, 11(8):R86, 2010.
- [49] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E.

- Stanley. Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.
- [50] P. Gope and T. Hwang. Bsn-care: a secure iot-based modern health-care system using body sensor network. *IEEE Sensors Journal*, 16(5):1368–1376, 2016.
- [51] A. Greco, G. Valenza, A. Lanata, E. P. Scilingo, and L. Citi. cvxEDA: A convex optimization approach to electrodermal activity processing. *IEEE Transactions on Biomedical Engineering*, 63(4):797–804, 2016.
- [52] M. A. Hanson, H. C. Powell Jr, A. T. Barth, K. Ringgenberg, B. H. Calhoun, J. H. Aylor, and J. Lach. Body area sensor networks: Challenges and opportunities. *Computer*, 42(1):58, 2009.
- [53] J. Hernandez, Y. Li, J. M. Rehg, and R. W. Picard. Bioglass: Physiological parameter estimation using a head-mounted wearable device. In *2014 EAI 4th International Conference on Wireless Mobile Communication and Healthcare*, pages 55–58. IEEE, 2014.
- [54] J. Hernandez, I. Riobo, A. Rozga, G. D. Abowd, and R. W. Picard. Using electrodermal activity to recognize ease of engagement in children during social interactions. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 307–317. ACM, 2014.
- [55] C.-Y. Huang, M.-C. Chan, C.-Y. Chen, and B.-S. Lin. Novel wearable and wireless ring-type pulse oximeter with multi-detectors. *Sensors*, 14(9):17586–17599, 2014.
- [56] R. E. Hunter. Skiing injuries. *The American journal of sports medicine*, 27(3):381–389, 1999.

- [57] S. Imani, A. J. Bandothkar, A. V. Mohan, R. Kumar, S. Yu, J. Wang, and P. P. Mercier. A wearable chemical-electrophysiological hybrid biosensing system for real-time health and fitness monitoring. *Nature communications*, 7, 2016.
- [58] N. Iyengar, C. Peng, R. Morin, A. L. Goldberger, and L. A. Lipsitz. Age-related alterations in the fractal scaling of cardiac inter-beat interval dynamics. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 271(4):R1078–R1084, 1996.
- [59] H. Jiang, G. Yang, X. Gui, N. Wu, and T. Zhang. Emotion recognition system design using multi-physiological signals. In *2012 IEEE 11th International Conference on Cognitive Informatics & Cognitive Computing*, pages 499–503. IEEE, 2012.
- [60] K. J. Joosen, J. Mesman, M. J. Bakermans-Kranenburg, S. Pieper, P. S. Zeskind, and M. H. van IJzendoorn. Physiological reactivity to infant crying and observed maternal sensitivity. *Infancy*, 18(3):414–431, 2013.
- [61] S. L. Joshi, R. A. Vatti, and R. V. Tornekar. A survey on ECG signal denoising techniques. In *2013 International Conference on Communication Systems and Network Technologies*, pages 60–64. IEEE, 2013.
- [62] P.-G. Jung, G. Lim, S. Kim, and K. Kong. A wearable gesture recognition device for detecting muscular activities based on air-pressure sensors. *IEEE Transactions on Industrial Informatics*, 11(2):485–494, 2015.
- [63] G. Jurman, S. Merler, A. Barla, S. Paoli, A. Galea, and C. Furlanello. Algebraic stability indicators for ranked lists in molecular profiling. *Bioinformatics*, 24(2):258–264, 2008.

- [64] G. Jurman, S. Riccadonna, and C. Furlanello. A comparison of MCC and CEN error measures in multi-class prediction. *PloS one*, 7(8):e41882, 2012.
- [65] G. Jurman, S. Riccadonna, R. Visintainer, and C. Furlanello. Algebraic comparison of partial lists in bioinformatics. *PloS one*, 7(5):e36540, 2012.
- [66] M. S. Koehle, R. Lloyd-Smith, and J. E. Taunton. Alpine ski injuries and their prevention. *Sports Medicine*, 32(12):785–793, 2002.
- [67] S. Koelstra, C. Muhl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras. Deap: A database for emotion analysis using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31, 2012.
- [68] S. D. Kreibig. Autonomic nervous system activity in emotion: A review. *Biological psychology*, 84(3):394–421, 2010.
- [69] S. Lapi, F. Lavorini, G. Borgioli, M. Calzolari, L. Masotti, M. Pistolesi, and G. A. Fontana. Respiratory rate assessments using a dual-accelerometer device. *Respiratory physiology & neurobiology*, 191:60–66, 2014.
- [70] R. W. Levenson. The autonomic nervous system and emotion. *Emotion Review*, 6(2):100–112, 2014.
- [71] Q. Li and G. D. Clifford. Dynamic time warping and machine learning for signal quality assessment of pulsatile signals. *Physiological measurement*, 33(9):1491, 2012.
- [72] H.-C. Lin and R. McFatter. Empathy and distress: two distinct but related emotions in response to infant crying. *Infant Behavior and Development*, 35(4):887–897, 2012.

- [73] J. C. Liu, S. Verhulst, S. A. Massar, and M. W. Chee. Sleep deprived and sweating it out: the effects of total sleep deprivation on skin conductance reactivity to psychosocial stress. *Sleep*, 38(1):155, 2015.
- [74] Y. Liu and O. Sourina. EEG-based valence level recognition for real-time applications. In *2012 International Conference on Cyberworlds*, pages 53–60. IEEE, 2012.
- [75] K. V. Madhav, M. R. Ram, E. H. Krishna, N. R. Komalla, and K. A. Reddy. Robust extraction of respiratory activity from ppg signals using modified mspca. *IEEE Transactions on Instrumentation and Measurement*, 62(5):1094–1106, 2013.
- [76] M. Malik. Heart rate variability. *Annals of Noninvasive Electrocardiology*, 1(2):151–181, 1996.
- [77] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schroder. The SEMAINE database: Annotated multimodal records of emotionally coloured conversations between a person and a limited agent. *IEEE Transactions on Affective Computing*, 3(1):5–17, 2012.
- [78] B. Mehler, B. Reimer, J. Coughlin, and J. Dusek. Impact of incremental increases in cognitive workload on physiological arousal and performance in young adult drivers. *Transportation Research Record: Journal of the Transportation Research Board*, 2138:6–12, 2009.
- [79] J. Mineraud, O. Mazhelis, X. Su, and S. Tarkoma. A gap analysis of internet-of-things platforms. *Computer Communications*, 89:5–16, 2016.
- [80] A. C. Miu, R. M. Heilman, and M. Miclea. Reduced heart rate variability and vagal tone in anxiety: trait versus state, and the effects of autogenic training. *Autonomic Neuroscience*, 145(1):99–103, 2009.

- [81] D. Mønster, D. D. Håkonsson, J. K. Eskildsen, and S. Wallot. Physiological evidence of interpersonal dynamics in a cooperative production task. *Physiology & behavior*, 156:24–34, 2016.
- [82] F. J. O. Morales and D. Roggen. Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pages 92–99. ACM, 2016.
- [83] A. Muaremi, J. Seiter, G. Tröster, and A. Bexheti. Monitor and understand pilgrims: Data collection using smartphones and wearable devices. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous computing*, pages 679–688. ACM, 2013.
- [84] S. Nemati, M. M. Ghassemi, V. Ambai, N. Isakadze, O. Levantsevych, A. Shah, and G. D. Clifford. Monitoring and detecting atrial fibrillation using wearable technology. In *2016 IEEE 38th Annual International Conference of the Engineering in Medicine and Biology Society*, pages 3394–3397. IEEE, 2016.
- [85] A. Pantelopoulos and N. G. Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):1–12, 2010.
- [86] B.-J. Park, E.-H. Jang, S.-H. Kim, C. Huh, and J.-H. Sohn. Seven emotion recognition by means of particle swarm optimization on physiological signals. In *2012 9th IEEE International Conference on Networking, Sensing and Control*, pages 277–282. IEEE, 2012.
- [87] M. T. Petterson, V. L. Begnoche, and J. M. Graybeal. The effect of motion on pulse oximetry and its clinical significance. *Anesthesia & Analgesia*, 105(6):S78–S84, 2007.

- [88] R. W. Picard, E. Vyzas, and J. Healey. Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE transactions on pattern analysis and machine intelligence*, 23(10):1175–1191, 2001.
- [89] M.-Z. Poh, T. Loddenkemper, C. Reinsberger, N. C. Swenson, S. Goyal, M. C. Sabtala, J. R. Madsen, and R. W. Picard. Convulsive seizure detection using a wrist-worn electrodermal activity and accelerometry biosensor. *Epilepsia*, 53(5):e93–e97, 2012.
- [90] A. Porta, S. Guzzetti, N. Montano, R. Furlan, M. Pagani, A. Malliani, and S. Cerutti. Entropy, entropy rate, and pattern classification as tools to typify complexity in short heart period variability series. *IEEE Transactions on Biomedical Engineering*, 48(11):1282–1291, 2001.
- [91] D. S. Quintana, A. J. Guastella, T. Outhred, I. B. Hickie, and A. H. Kemp. Heart rate variability is associated with emotion recognition: Direct evidence for a relationship between the autonomic nervous system and social cognition. *International Journal of Psychophysiology*, 86(2):168–172, 2012.
- [92] N. M. Rad, A. Bizzego, S. M. Kia, G. Jurman, P. Venuti, and C. Furlanello. Convolutional neural network for stereotypical motor movement detection in autism. *arXiv preprint arXiv:1511.01865*, 2015.
- [93] D. S. Raju, M. S. Manikandan, and R. Barathram. An automated method for detecting systolic peaks from arterial blood pressure signals. In *2014 IEEE Students' Technology Symposium*, pages 41–46. IEEE, 2014.

- [94] R. M. Ram, K. V. Madhav, E. H. Krishna, N. R. Komalla, K. Sivani, and K. A. Reddy. Dual-tree complex wavelet transform for motion artifact reduction of PPG signals. In *Medical Measurements and Applications Proceedings (MeMeA), 2012 IEEE International Symposium on*, pages 1–4. IEEE, 2012.
- [95] R. M. Ram, K. V. Madhav, E. H. Krishna, N. R. Komalla, K. Sivani, and K. A. Reddy. ICA-based improved DTCWT technique for MA reduction in PPG signals with restored respiratory information. *IEEE Transactions on Instrumentation and Measurement*, 62(10):2639–2651, 2013.
- [96] P. P. Ray. A survey of IoT cloud platforms. *Future Computing and Informatics Journal*, 2017.
- [97] G. Regalia, F. Onorati, M. Migliorini, and R. Picard. An improved wrist-worn convulsive seizure detector based on accelerometry and electrodermal activity sensors. In *American Epilepsy Society annual meeting 2015*, 2015.
- [98] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.
- [99] R. Richer, P. Blank, D. Schuldhaus, and B. M. Eskofier. Real-time ECG and EMG analysis for biking using Android-based mobile devices. In *Wearable and Implantable Body Sensor Networks (BSN), 2014 11th International Conference on*, pages 104–108. IEEE, 2014.
- [100] L. Rodríguez-Liñares, M. Lado, X. Vila, A. Méndez, and P. Cuesta. gHRV: Heart rate variability analysis made easy. *Computer methods and programs in biomedicine*, 116(1):26–38, 2014.

- [101] L. Rodríguez-Liñares, X. Vila, A. Mendez, M. Lado, and D. Olivieri. RHRV: An R-based software package for heart rate variability analysis of ECG recordings. In *3rd Iberian Conference in Systems and Information Technologies*, pages 565–574, 2008.
- [102] T. Roh, K. Bong, S. Hong, H. Cho, and H.-J. Yoo. Wearable mental-health monitoring platform with independent component analysis and nonlinear chaotic analysis. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4541–4544. IEEE, 2012.
- [103] L. E. Romero, P. Chatterjee, and R. L. Armentano. An iot approach for integration of computational intelligence and wearable sensors for parkinson’s disease diagnosis and monitoring. *Health and Technology*, pages 1–6, 2016.
- [104] W. T. Roth, M. E. Dawson, and D. L. Filion. Publication recommendations for electrodermal measurements. *Psychophysiology*, 49:1017–1034, 2012.
- [105] G. Ruedl, A. Schranz, C. Fink, E. Pocecco, W. Nachbauer, and M. Burtscher. Are acl injuries related to perceived fatigue in female skiers? In *Skiing Trauma and Safety, 18th Volume*. ASTM International, 2011.
- [106] N. Ruiz-Robledillo and L. Moya-Albiol. Lower electrodermal activity to acute stress in caregivers of people with autism spectrum disorder: an adaptive habituation to stress. *Journal of autism and developmental disorders*, 45(2):576–588, 2015.
- [107] A. Sano and R. W. Picard. Stress recognition using wearable sensors and mobile phones. In *2013 Humaine Association Conference on Af-*

- fective Computing and Intelligent Interaction*, pages 671–676. IEEE, 2013.
- [108] A. Sano, R. W. Picard, and R. Stickgold. Quantitative analysis of wrist electrodermal activity during sleep. *International Journal of Psychophysiology*, 94(3):382–389, 2014.
- [109] A. Schäfer and J. Vagedes. How accurate is pulse rate variability as an estimate of heart rate variability?: A review on studies comparing photoplethysmographic technology with an electrocardiogram. *International journal of cardiology*, 166(1):15–29, 2013.
- [110] SEQC/MAQC-III Consortium. A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. *Nature biotechnology*, 32(9):903–914, 2014.
- [111] C. Setz, B. Arnrich, J. Schumm, R. La Marca, G. Tröster, and U. Ehlert. Discriminating stress from cognitive load using a wearable EDA device. *IEEE Transactions on information technology in biomedicine*, 14(2):410–417, 2010.
- [112] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *IEEE network*, 18(4):45–50, 2004.
- [113] M. Soleymani, J. J. Kierkels, G. Chanel, and T. Pun. A bayesian framework for video affective representation. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pages 1–7. IEEE, 2009.
- [114] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic. A multimodal database for affect recognition and implicit tagging. *IEEE Transactions on Affective Computing*, 3(1):42–55, 2012.

- [115] N. Spiro and T. Himberg. Analysing change in music therapy interactions of children with communication difficulties. *Philosophical Transaction of the Royal Society B*, 371:20150374, 2016.
- [116] S. M. Srinivasan, M. Kaur, I. K. Park, T. D. Gifford, K. L. Marsh, and A. N. Bhat. The effects of rhythm and robotic interventions on the imitation/praxis, interpersonal synchrony, and motor performance of children with autism spectrum disorder (ASD): a pilot randomized controlled trial. *Autism research and treatment*, 2015.
- [117] J. X. Sun, A. T. Reisner, M. Saeed, and R. G. Mark. Estimating cardiac output from arterial blood pressure waveforms: a critical evaluation using the MIMIC II database. In *Computers in Cardiology, 2005*, pages 295–298. IEEE, 2005.
- [118] M. Swan. Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0. *Journal of Sensor and Actuator Networks*, 1(3):217–253, 2012.
- [119] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida. Wearable photoplethysmographic sensors—past and present. *Electronics*, 3(2):282–302, 2014.
- [120] M. P. Tarvainen, J.-P. Niskanen, J. A. Lipponen, P. O. Ranta-Aho, and P. A. Karjalainen. Kubios HRV—heart rate variability analysis software. *Computer methods and programs in biomedicine*, 113(1):210–220, 2014.
- [121] K. Tomazin, A. Dolenc, and V. Strojnik. High-frequency fatigue after alpine slalom skiing. *European Journal of Applied Physiology*, 103(2):189–194, 2008.

- [122] C. A. Torres-Valencia, H. F. García-Arias, M. A. A. López, and A. A. Orozco-Gutiérrez. Comparative analysis of physiological signals and electroencephalogram (EEG) for multimodal emotion recognition using generative models. In *2014 XIX Symposium on Image, Signal Processing and Artificial Vision*, pages 1–5. IEEE, 2014.
- [123] A. Uusitalo, T. Mets, K. Martinmäki, S. Mauno, U. Kinnunen, and H. Rusko. Heart rate variability related to effort at work. *Applied ergonomics*, 42(6):830–838, 2011.
- [124] G. Valenza, A. Lanatá, and E. P. Scilingo. Improving emotion recognition systems by embedding cardiorespiratory coupling. *Physiological measurement*, 34(4):449, 2013.
- [125] M. van Dooren, J. J. G. de Vries, and J. H. Janssen. Emotional sweating across the body: Comparing 16 different skin conductance measurement locations. *Physiology & behavior*, 106(2):298–304, 2012.
- [126] K. M. Warren, J. R. Harvey, K. H. Chon, and Y. Mendelson. Improving pulse rate measurements during random motion using a wearable multichannel reflectance photoplethysmograph. *Sensors*, 16(3):342, 2016.
- [127] G. E. White and G. D. Wells. The effect of on-hill active recovery performed between runs on blood lactate concentration and fatigue in alpine ski racers. *The Journal of Strength & Conditioning Research*, 29(3):800–806, 2015.
- [128] World Medical Association. World Medical Association Declaration of Helsinki: ethical principles for medical research involving human subjects. *Journal of the American Medical Association*, 310(20):2191, 2013.

- [129] S. Yadhuraj and H. Harsha. Motion artifact reduction in photoplethysmographic signals: A review. *International Journal of Innovative Research and Development*, 2(3):626–640, 2013.
- [130] K. Yan, B. Tracie, M. Marie-Ève, H. Mélanie, B. Jean-Luc, T. Benoit, M. St-Onge, and L. Marie. Innovation through wearable sensors to collect real-life data among pediatric patients with cardiometabolic risk factors. *International Journal of Pediatrics*, 2014.



# Appendix A

## DBD: Derivative Based Detection of heart beats

In this Chapter we introduce and validate a pipeline for Heart Rate Variability (HRV) analysis on signals acquired with Wearable Devices (WDs). Currently two main sensing technologies are available to acquire the cardiac activity with WDs: electrocardiography or photoplethysmography (PPG) sensors. The former is used to acquire the electrocardiogram (ECG), while the PPG sensors detect the Blood Volume Pulse (BVP); see 2.1.1 for a description of sensing technologies for cardiac signals.

A typical HRV analysis pipeline (see Figure 3.1) is composed of three stages [76]: (a) preprocessing; (b) beat detection and (c) computation of indicators, Typically applied on ECG signals. However BVP signals can also be used [109]: from the signal processing point of view the main difference is in the second stage, where, due to the different waveform of BVP compared to ECG, specific algorithms have been defined to correctly estimate the beat position.

### Preprocessing

The preprocessing part aims at increasing the signal-noise ratio (SNR) by filtering, detrending and removal of undesired components. In particular,

several algorithms and technical solutions have been adopted to deal with moving artifacts (see, for instance [42, 94, 95, 126, 129] for a review), but complete removal is still an open issue due to the impulsive and time-dependency characteristic of body movements.

As an alternative, it has been proposed to estimate the amount of noise and reject those portions where this is too high [126, 84, 71]. However, in this chapter we prefer do not discuss the denoising, focusing instead on the second stage of the pipeline: detection of heart beats and then computation of Inter Beat Intervals (IBIs).

## **Beat detection**

The detection of the beat position in an ECG signal corresponds to the identification of the R peak, which is usually well recognizable due to its higher amplitude and characteristic impulsive shape. Peaks can be identified using different algorithms (see for instance [61]) also for real-time processing (see [33]); however identification accuracy is weakened by high frequency noise and trends that should have been removed in the preprocessing stage.

A BVP signal does not provide a wave portion corresponding to the R peak of the ECG signal; in addition, beat detection is made difficult also by the variability due to posture or even individual physiological characteristics [109]. Indeed, a different family of algorithms is used to identify the correct beat position from PPG [1, 117, 93]. In general, it emerges that a reliable and general method is still missing.

Mostly, such algorithms are usually proposed for use in diagnostics and health care; in the latter case, the interest is a continuous HR monitoring and thus a rougher measure is adopted compared to HRV analysis (heart rate in beats per minutes vs measuring distance between beats at higher resolution of milliseconds). Further, only few algorithms have been pro-

posed appositely for analysis of signals acquired with WDs. They present usually even higher variability in terms of waveform shape and thus algorithms need to be robust to time-variant characteristics of the signal and variations of amplitude with these sensors.

## Computation of HRV indicators

The first step in order to compute HRV indicators, is detecting heart beats from the cardiac signal, then derive the distance between consecutive beats (IBI). HRV indicators are used to describe a particular aspect of the heart physiology or activity of the autonomic system (for a complete review see [76]).

Here we propose a signal processing pipeline suited for signals acquired with WDs, which could serve as a reference methodology for future papers and comparison of new algorithms (Section A.1). In particular, we describe the Derivative Based Detection (DBD) algorithm for beat detection on BVP signals and the Reverse Combinatorial Optimization (RCO) for correction of misdetection errors. We compare the DBD with the automatic beat detection algorithm [1] showing that DBD is more robust and appropriate for signals acquired with WDs and correction by RCO further improves the results. Section A.2 describes the two datasets we used to test the pipeline: a) the Fantasia dataset [58] and b) our WCS dataset, designed to compare signals from clinical (Thought Technology FlexComp Infinity) and wearable devices (Empatica E4 and ComfTech HeartBand). We present and discuss the result in terms of beat detection errors and differences in the computed HRV indicators (Section A.3 and A.4).

## A.1 Beat detection pipeline

In this section we present the pipeline for beat detection on cardiac signals to extract the IBI signal. The pipeline addresses the issues originated by use of acquired wearable technologies, however it can be applied on signals from medical-grade devices, where it is expected to give even better performances (see Section 5).

In particular, we present the DBD algorithm for estimation of beat position in BVP signals and the RCO algorithm for errors detection and correction. The interest of the DBD algorithm is that most of WDs embed a PPG sensor to acquire cardiac signals. We omit to discuss the preprocessing step (for instance band-pass filtering to remove trends and high frequency noise) as already extensively discussed in literature.

### A.1.1 Oversampling

Effects of lower sampling frequency on the HRV indicators can be mitigated by signal interpolation [36]. The first step in the pipeline is therefore an oversampling to decrease the error in the estimation of the beat position. We apply a cubic spline interpolation, with an output sampling frequency of at least 1000 Hz to estimate distances between beats with 1 ms resolution.

### A.1.2 Energy estimation

Several methods have been proposed to deal with artifacts associated to body movements but in case of high dynamic movements or severe signal corruption (for instance due to electrodes disconnection or displacement) they might be ineffective [126]. Further, the presence of artifacts in the signal will always impact on the error of estimation of the beat position. For this reason, beside addressing the removal of artifacts, it is important to obtain a measure of reliability of the estimates. This becomes fundamental

in online applications (e.g. when short-term or real-time indicator triggers a decision logic), or when processing large amounts of streaming flows which would make practically unfeasible to manually check the results of beat position estimation.

To this aim we propose to use the Energy of the Signal Derivative:

$$e_n = \frac{\sum_{n=2}^N (x_n - x_{n-1})^2}{N - 1} \quad (\text{A.1})$$

where  $x_k$  is the value of  $k$ -th sample of the signal and  $N$  is the total number of samples.

To consider for local variations of the noise the energy (A.1) should be iteratively computed within a windowing process that runs along the whole signal:

$$y_k = f(x_{kw_s}, \dots, x_{kw_s+w_l}) \quad (\text{A.2})$$

where  $k$  is the current windowing iteration  $w_s$  is the number of samples the window shifts at each iteration and  $w_l$  is the number of samples of the window.

In absence of artifacts,  $e_n$  depends only on the dynamic of the signal which can be considered approximately time invariant. When artifacts corrupt the signal their contribution will increase the value of  $e_n$  which can therefore be used to estimate the presence and magnitude of artifacts. A threshold value should be defined to identify the noisy portion of signals that would provide an unreliable estimation of the beat position and, consequently, of the computed HRV indicator.

Note that, when the wearable devices embeds also an accelerometer,  $e_n$  can be computed from the acceleration signal to provide a more reliable quantification of artifacts due to body movement.

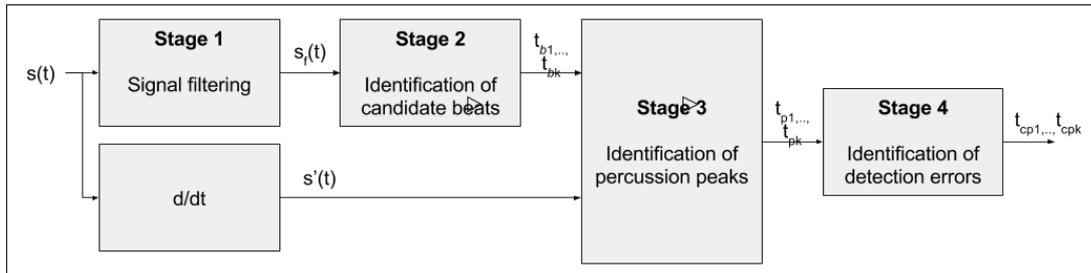


Figure A.1: Overview of the DBD algorithm with the four processing stages.

### A.1.3 The DBD algorithm

The DBD algorithm estimates the position of the percussion peak in signals acquired with an optical sensor and it is designed to be robust when applied on signals from WDs which present high variability of waveform shapes and amplitudes.

As for other algorithms proposed in literature (e.g. [1, 93]), the DBD algorithm adopts a sequence of processing steps. We use three main stages, with an additional stage for the automatic identification of detection errors (RCO). An overview of the DBD procedure is presented in Figure A.1.

The algorithm is regulated by the parameter  $f_{\max}$  which is the expected maximal heartbeat frequency. We empirically set  $f_{\max}$  to 2 Hz, corresponding to 120 beats per minute). However, the parameter can be changed accordingly to physiological condition (e.g. sleep) or age of individual (e.g. infant, newborn). In the following subsections a detailed description for each stage of the algorithm is given.

#### DBD Stage 1: Signal filtering

The first stage aims at filtering the original signal to extract the approximate beat positions. This information will be used in the following steps to target the detection of the percussion peak. We use a low pass infinite impulse response filter ( $f_{pass} = 1.2f_{\max}$ ,  $f_{stop} = 3f_{\max}$ ) to filter out high frequency components of the signal.

**DBD Stage 2: Identification of candidate beat position**

On the filtered signal from stage 1 we identify the candidate beats by adaptive peak detection. First the local range of the signal is estimated by windowing (window width:  $1.5(f_{samp})/f_{max}$ , window shift:  $f_{samp}/f_{max}$ ) and evaluating the range for each window, then interpolating according to the original sampling frequency of the signal. The local range of the signal is used to modulate the peak detection: a local maximum  $x_{p_i}$  is considered valid if the difference with the following local minimum is greater than half of the range of the signal at the local minimum.

**DBD Stage 3: Identification of percussion peak position**

For each peak instant detected in stage 2 ( $t_{p_i}$ ) we extract the 250 ms before ( $x[t_{p_i} - 0.25, t_{p_i}]$ ) which is the candidate portion ( $C_{p_i}(t)$ ) where to locate the percussion peak. We compute the derivative of the candidate portion  $C'_{p_i}(t)$  where we detect the maximum  $t_{m_i}$  corresponding to the steepest instant of the rising part of the pulse and the instant  $t_{0_i}$  following  $t_{m_i}$ , corresponding to the first local minimum of  $|C'_{p_i}(t)|$ .

$C'_{p_i}(t_{0_i})$  is expected to be equal to 0 and is considered the instant of the percussion peak. Figure A.2 shows the steps to identify  $t_{0_i}$  on a beat pulse. The IBI signal is then computed as difference between consecutive percussion instants.

**A.1.4 The RCO algorithm**

The RCO algorithm aims at identifying and correcting errors in the IBI signal in particular when the correct beat position is not detected because of a false peak before the true peak. It depends only on the IBI signal and thus it can be employed also to correct beats detected on ECG signals.

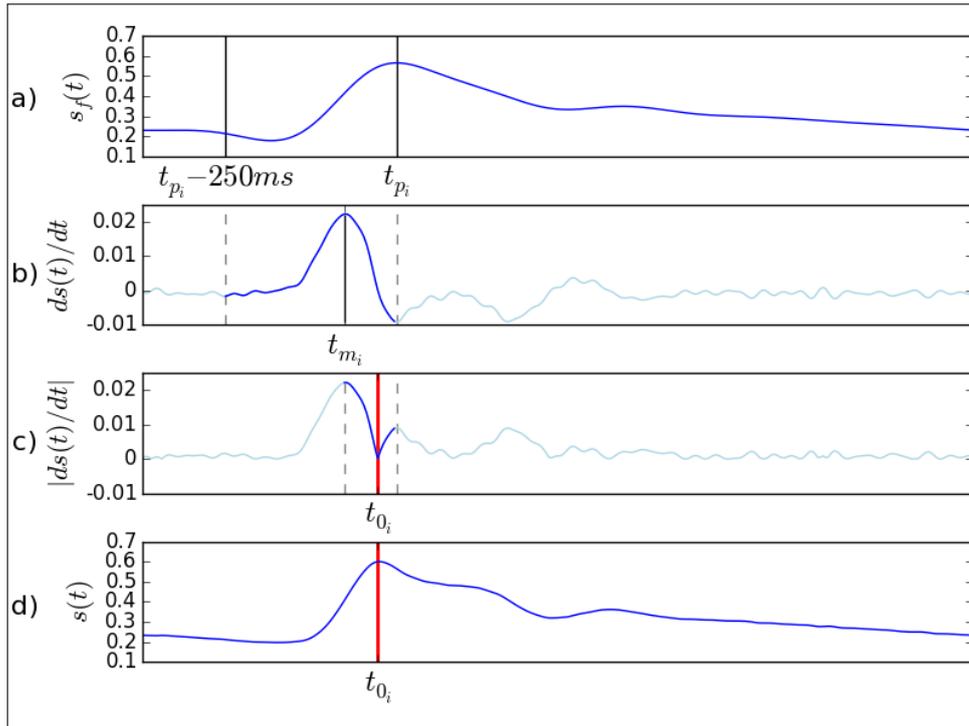


Figure A.2: Steps of stage 3 to identify  $t_{0_i}$ . From top to bottom: a) identification of the maximum  $t_{p_i}$  and  $t_{p_i} - 0.25$  on the filtered signal  $x_f(t)$ ; b) identification of the maximum of the derivative of the signal  $x_n - x_{n-1}$  in the interval  $t_{p_i} - 0.25, t_{p_i}$ ; c) identification of the local minimum of  $|ds(t)/dt|$  in the interval  $[t_{m_i}, t_{p_i}]$ , corresponding to  $t_{0_i}$ , the percussion peak in the original signal (d).

### Adaptive Outlier Detection

The basic algorithm of the RCO is the Adaptive Outlier Detection (AOD). The AOD uses a fixed size cache vector  $IBI_c = (ibi_1, ibi_2, \dots, ibi_k)$  to store the last  $k$  valid IBI values, in order to adapt to IBI variability. The size  $k$  is empirically set to 5 and the cache is initialized with the median value of the IBI signal. Outliers detection is also regulated by the sensitivity parameter  $\phi$ , which is empirically set to 0.25. However, smaller values of  $\phi$  and greater values of  $k$  improve precision.

A detected beat is considered valid if its corresponding IBI value is within the interval  $[(1-\phi)ibi_{median}, (1+\phi)ibi_{median}]$ , where  $ibi_{median}$  is the median of the values in  $IBI_c$ . When a new valid beat is detected, its IBI value is used to update  $IBI_c$  using the First-In-First-Out rule.  $IBI_c$  is re-initialized when  $k$  consecutive non valid beats are detected.

### RCO Stage 1: Detection of questionable beats

The first step of the RCO is the application of AOD on the IBI in both forward and backward direction: those beats that are detected as false positives in both the directions are rejected and those that are detected as true positives in both directions are validated. Each remaining detected beat from backward direction is paired to the nearest detected beat from forward direction. When the distance is less than 250 ms, the pair is set to be checked with the combinatorial optimization.

### RCO Stage 2: Combinatorial correction

Segments are generated for each sequence of one or more consecutive pairs, by concatenating the last valid beat before the sequence and the first valid beat after the sequence. Then all the combinations of beats are generated by selecting alternatively the beat from the forward or from the backward

direction. For each combination we compute the variability index, defined as follows:

$$V = \sum_2^N |IBI_n - IBI_{n-1}| \quad (\text{A.3})$$

Then the combination with the minimum value of  $V$  is selected as the best estimate of beats in the sequence. In the final RCO step, the AOD algorithm is run forward on the corrected IBI to identify possible outliers on IBI series that might have not been corrected.

## A.2 Materials and methods

To validate the proposed pipeline we first aimed at testing the performances of the DBD algorithm on a reference database. As Physionet<sup>1</sup> [49] does not provide a dataset of BVP signals with beats annotations, we selected the *Fantasia* dataset [58] which contains the ECG and Blood Pressure (BP) signals (sampling rate: 250 Hz) of twenty subjects (10 young, mean age=27yr and 10 elderly, mean age=74yr) together with beat annotations. Note, however, that BP is a different physiological measure than BVP, though similar. The signals were acquired while the subjects were watching the movie “Fantasia” (Disney, 1940) laying supine and maintaining an inactive state. The heartbeats were first automatically annotated on the ECG and then verified by visual inspection to provide the reference beats series of the *Fantasia* dataset.

In order to validate the pipeline on signals from WDs, we tested the whole pipeline on the WCS dataset (see Chapter 4). In particular, we applied the pipeline to the BVP signals from the Empatica E4 and from the FlexComp to compare results between a wearable and a medical-grade device.

---

<sup>1</sup><http://www.physionet.org>

### A.2.1 Metrics

We defined two different groups of metrics to quantify performance of the beat detection step and to assess reliability of HRV indicators. The first group compares the results of the beat detection step, the second compares the values of HRV indicators. Four algorithms for beat detection are used: DBD, DBD+RCO, the algorithm for beat detection proposed in [1] with (ABOY+CORR) and without (ABOY) the beat correction procedure.

#### Beat detection metrics

The results of the beat detection step are validated on the reference beat series first by counting the number of true positives (TPs), false positives (FPs) and false negatives (FNs). To compute the number of TPs, we first try to pair each beat  $b_{det}$  of the detected beat series to a beat  $b_{ref}$  of the reference beats series. The pairing is considered valid when the distance between  $b_{det}$  and  $b_{ref}$  is below 0.5 s. The remaining unpaired detected beats are considered as FPs and the remaining unpaired reference beats are considered as FNs. Then we counted the number of TPs ( $n_{TP}$ ), the number of FPs ( $n_{FP}$ ) and the number of FNs ( $n_{FN}$ ) to compute the overall *recall* ( $(n_{TP})/(n_{TP} + n_{FN})$ ) and *precision* ( $(n_{TP})/(n_{TP} + n_{FP})$ ).

We also compared the reference IBI series  $IBI_{ref}$  computed from the reference beats series with IBI values resulting from the detected beats  $IBI_{est}$ . We consider the overall root mean square error (rmse) between the interpolated version of  $IBI_{est}$  and  $IBI_{ref}$ . Both the IBI series are interpolated with cubic spline at 4 Hz.

#### HRV reliability metrics

The second category analyses the reliability of computed HRV indicators. Ten 30-seconds portions of the IBI series are selected and 4 HRV indi-

cators are computed for each portion. Two indicators are chosen from the time-domain category: mean of IBI (RRmean), root mean square of the differences of subsequent IBI (RMSSD), and two from the frequency-domain category: power in the low frequency band [0.04-0.15 Hz] (LF), power in the high frequency band [0.15-0.4 Hz] (HF). The HRV values obtained from the  $IBI_{ref}$  and from the  $IBI_{est}$  are compared using the Bland-Altman (BA) ratio as proposed in [109].

### Metrics estimation

The computation of metrics is performed by windowing (length: 30 s, shift: 10 s) along the whole signal, to obtain the distribution of values of each metric for each subject. Beat detection metrics are averaged to obtain the mean value for each subject, while the HRV values are used to compute the BA ratio:

$$r_{BA} = \frac{IC_d}{mean_m} \quad (\text{A.4})$$

where  $IC_d$  is the 95% confidence interval of the difference between the reference and estimated indicator value ( $d = (x_i^{ref} - x_i)$ ), and  $mean_m$  is the mean of the average between the reference and estimated indicator value ( $m = (x_i^{ref} + x_i)/2$ ). BA ratio is used to evaluate similarity between two sets of measurements. In general, it is used to compare a novel method of measurement with a reference. Two methods are considered as coincident when the BA ratio is below 0.1. A value between 0.1 and 0.2 corresponds to a fair similarity; when the BA ratio is above 0.2, the novel method is considered unreliable.

## A.3 Results

In the following subsections we discuss the results on the *Fantasia* dataset and the comparison between wearable and clinical devices.

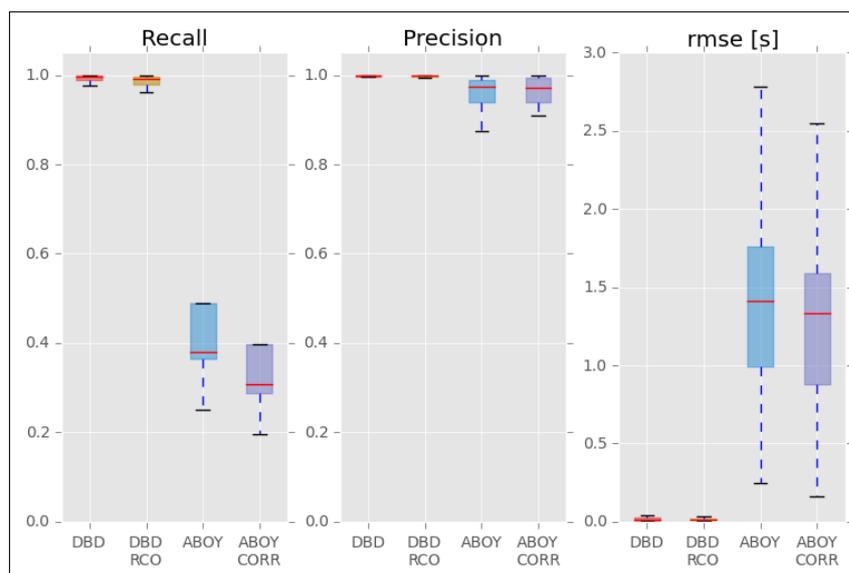


Figure A.3: Performances of IBI detection on the Fantasia dataset.

### A.3.1 Results on Fantasia dataset

For each subject we selected a 1000 s length portion of the BP signal which showed no sensor disconnections. Performance of IBI detection (see Figure A.3) appears similar for the DBD and the DBD-RCO pipelines. Performance of ABOY and ABOY-CORR is lower, both in terms of Recall and rmse. Similar results are found also for the Bland-Altman ratios of HRV indicators: DBD and DBD+RCO provide similar results, ABOY and ABOY-CORR are also comparable but value of ratios is higher. However, in general the distribution of the BA values is mostly above the 0.2 value; only for RRmean the ratio is below 0.1 for DBD and DBD-RCO and around 0.2 for ABOY and ABOY-CORR.

### A.3.2 Results on WCS dataset

For each subject we considered both baseline and movement portions to tackle the effect of artifacts. We applied the proposed pipeline to the three cardiac signals acquired during the experiment: (a) BVP signal from the

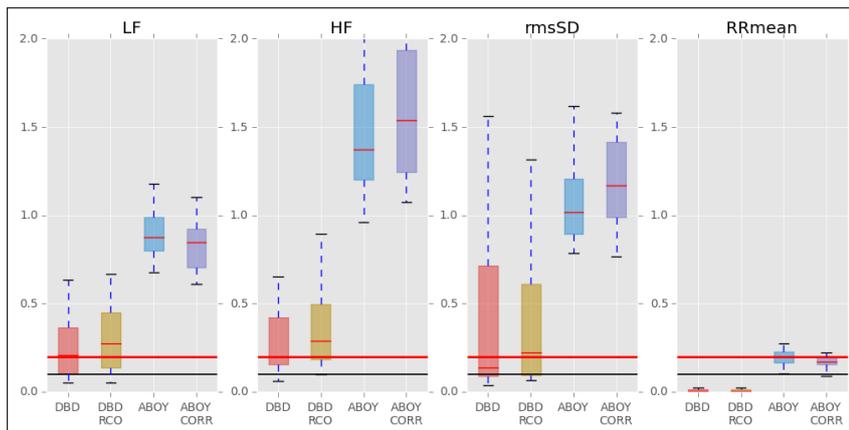


Figure A.4: BA ratio of HRV indicators on the Fantasia dataset.

FlexComp, (b) BVP signal from the Empatica E4 and c) the ECG signal from the Comftech HeartBand.

The ECG signal from the FlexComp is used to provide the ground truth: IBI were automatically detected and manually corrected by visual inspection to remove wrong detected peaks and add missing beats.

As with the Fantasia dataset, for the BVP signal (see Figure A.5) with the FlexComp and E4 on the baseline the proposed algorithms (DBD and DBD-RCO) perform better than the reference algorithms (ABOY and ABOY-CORR). In general presence of moving artifacts causes an increase in the rmse and decrease of Recall, with more evident effects on the E4.

Reliability indexes of HRV indicators (see Figure A.6) extracted from the FlexComp BVP are similar to what found for the Fantasia dataset: better performances of DBD and DBD-RCO algorithms but only RRmean with distribution of BA ratios under 0.2 (around 0 for DBD and DBD-RCO). Results on the BVP from E4 are comparable, although performances of DBD and DBD-RCO are lower.

Body movements cause all the ratios to increase. Again, RRmean from FlexCom appears a reliable indicator, although only the DBD-RCO algorithm presents values below 0.1. For the E4, instead, BA ratio of the

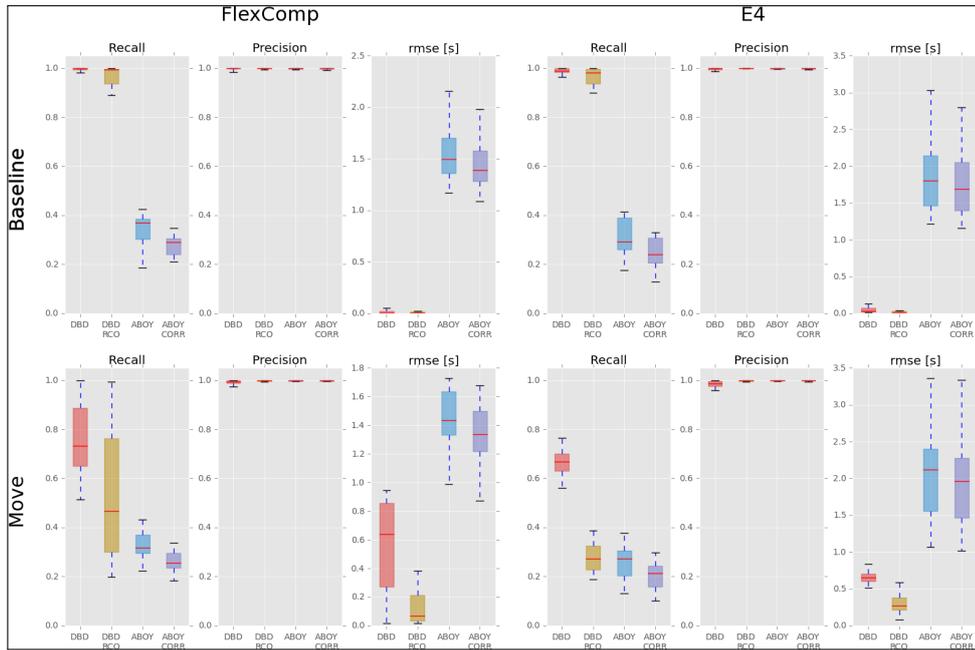


Figure A.5: Performances of IBI detection on the BVP from the FlexComp (left) and E4 (right) during baseline (top) and move (bottom) portions.

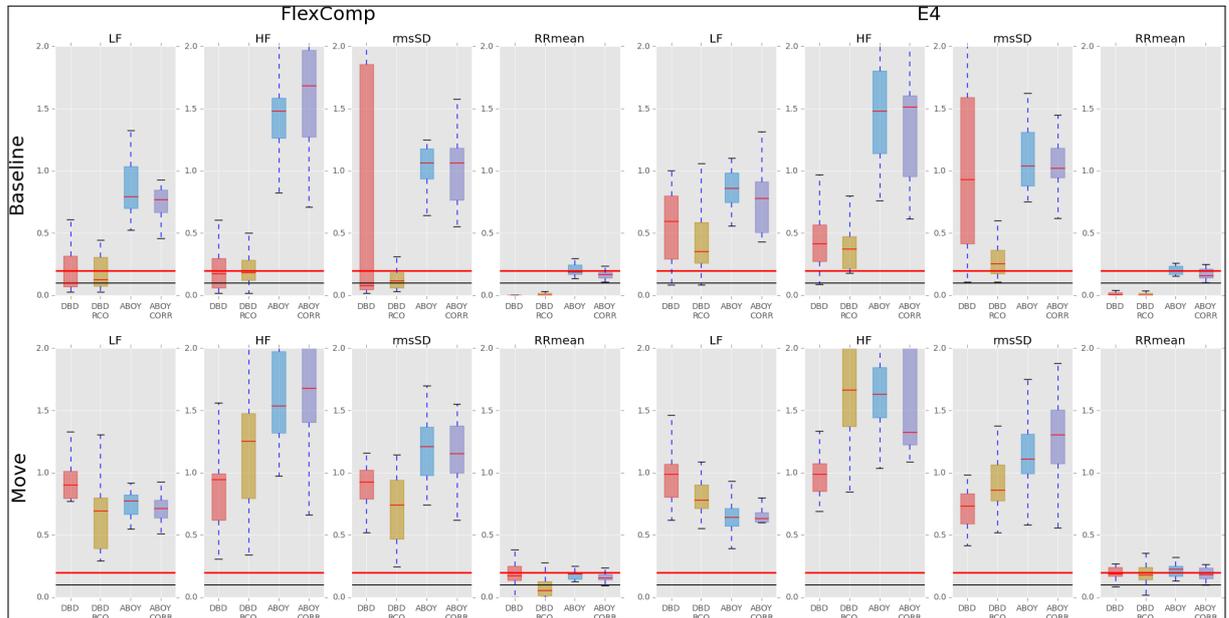


Figure A.6: BA ratio of HRV indicators on the BVP from the FlexComp (left) and E4 (right) during baseline (top) and move (bottom) portions. Vertical scale is limited to the interval [0,2] for clarity.

RRmean is around 0.2 for all algorithms.

## A.4 Discussion

In this chapter we proposed the DBD and the RCO a pipeline for detection and processing of cardiac signals from WDs, in particular for beat detection on signals from PPG sensors which represent a common embedded technology to monitor cardiac activity. We compared the proposed algorithm to a reference algorithm [1] in terms of beat detection, differences in the IBI series and HRV indicators, using an existing dataset of clinical signals and a dataset appositely created to assess performances on signals from WDs.

Performances of DBD+RCO improved respect to the reference algorithm in terms of beat detection and derived IBI values for both the types of devices (clinical and wearable) and both the experimental conditions (baseline and movement). The RCO also improves the results of the beat detection on the ECG signal from the WD Comftech HeartBand.

Performance of DBD+RCO improved with respect to the reference algorithm in terms of beat detection and derived IBI values for both the types of devices (clinical and wearable) and both the experimental conditions (baseline and movement). The RCO also improves the results of the beat detection on the ECG signal from the WD Comftech HeartBand.

In absence of body movements the three signals provided similar results in terms of beat detection and IBI series. Regarding HRV indicators the results confirmed that the BVP signals provide unreliable estimates, in particular for frequency domain indicators, as found in other studies [109]. The indicators computed on the Comftech HeartBand ECG signal showed high accuracy in terms of Bland-Altman ratio, however frequency domain indicators presented higher value, but still below the threshold value of 0.2.

During the movements portion the BVP signals from both the FlexComp and the Empatica E4 resulted highly affected by moving artifacts, with slightly better performances for the clinical version (FlexComp). All the HRV indicators, except the RRmean, presented high BA ratio, showing low reliability. Instead, the Comftech HeartBand showed good results in terms of beat detection and IBI series, while, among HRV indicators, only RRmean showed good accordance with the reference.

In summary, the proposed pipeline for processing of cardiac data improves the detection of beats on BVP. In general, cardiac signals acquired with WDs result appropriate to estimate IBI and RRmean, but the errors introduced by the technological constraints of these devices allow no accurate analysis of other HRV indicators. This study highlights also that in presence of body movements an electrocardiography-based device should be used instead of PPG sensors (both clinical and wearable) as it provides a signal which is less sensitive to moving artifacts.



## Appendix B

# Peak based estimation of Bateman parameters and of phasic skin response

The analysis of Electrodermal Activity (EDA) is used to investigate the psycho-physiological response in several research fields such as autism [106], sleep [73], and epilepsy [89].

The EDA signal originates by the variation of skin conductivity due to presence of sweat and it can be measured, for instance, by application of a constant voltage or current between two electrodes that are applied to the skin [104].

Two components contribute to the EDA signal: the Phasic Skin Response (PSR) and the Tonic Skin Response (TSR), which represent two different physiological phenomena [104]. The first depends on the triggering activity of the Sympathetic Nervous System (SNS) which sends a series impulses to the sweating glands through the sudomotor nerve. Each impulse causes an immediate and temporary increasing in the sweating level. Then, if no more impulses are sent, the sweat level decreases due to the diffusion on the skin. This physiological phenomenon is observed in the EDA signal as a sudden peak in the conductivity and slow recovery to the

baseline level. The PSR component is composed of the series of the peaks originated from the triggering of the SNS and their subsequent recovery.

The PSR component is overlapped to the TSR component which represents the long-term variations in the sweat level due to other physiological regulations: for instance thermoregulation and perspiration [14]. Thus, the importance of EDA to study the Autonomic Nervous System (ANS) activity is mainly associated to the analysis of the PSR component which needs to be discriminated by application of appropriate signal processing techniques.

The main goal is the identification of the peaks, as often subsequent peaks are masked by the slow recovering phase of the previous ones. Several approaches have been proposed, mainly based on deconvolution with a parametrized function [14, 4, 6]. In this study we refer to the approach of Benedek and colleagues [13]: they model the SNS triggering activity with a series of delta impulses and the sweating activity of the glands as a convolutional filter characterized by a Bateman function as Impulse Response Function (IRF). The reconstruction of the SNS triggering activity is performed in two stages: (a) the deconvolution of the EDA signal using the Bateman function as IRF and (b) the identification of inter peaks periods to identify the TSR and subsequently the PSR.

However, due to inter-subject variability the two parameters regulating the Bateman function can not be known *a-priori* and need to be estimated. To this aim, Benedek and colleagues proposed an optimization procedure based on a loss function which takes into account the amount of negative portions and average width of the delta impulses in the PSR.

In this work we present a novel pipeline which aims at improving the discrimination of PSR and TSR components by two key steps: (a) stochastic optimization of a peak-based loss function to estimate the Bateman parameters (Section B.1); (b) identification of inter-peak intervals based on

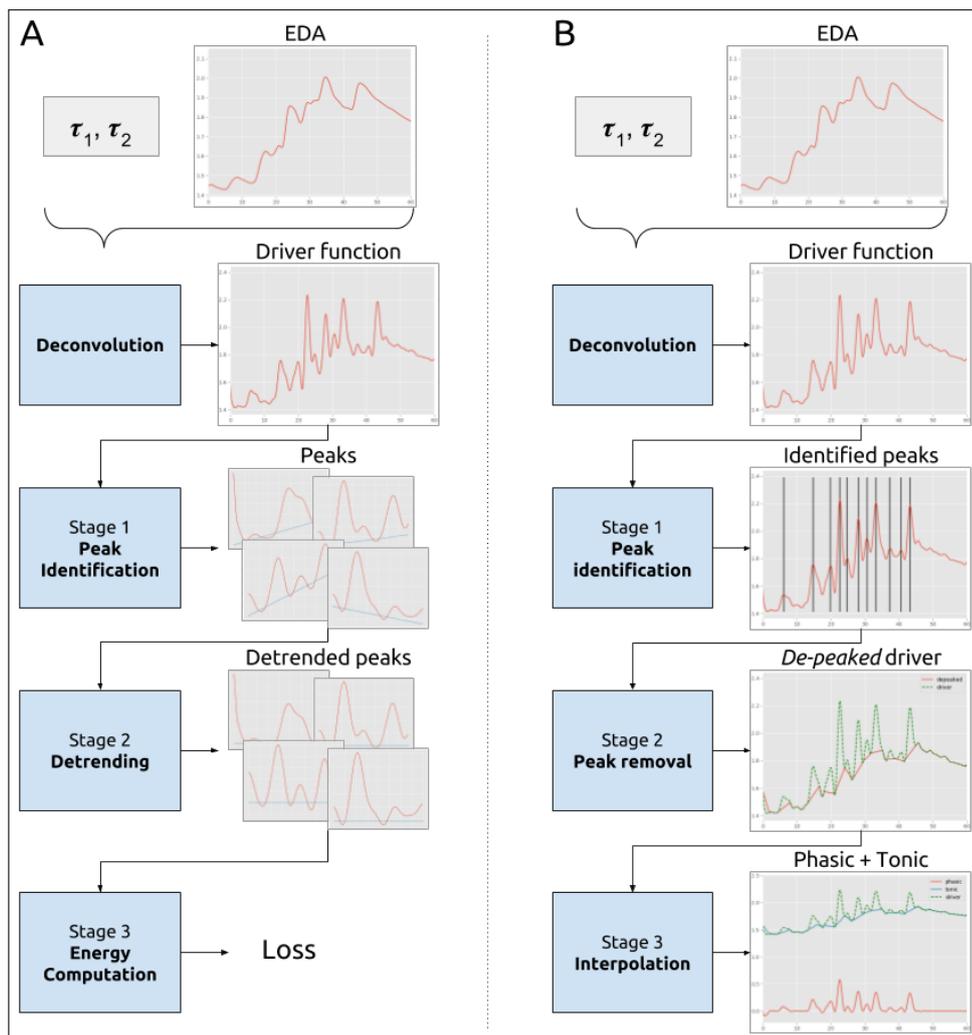


Figure B.1: Flow charts of the two novel algorithms introduced in this paper: A) peak based loss function and B) derivative-based phasic estimation.

signal derivative to estimate the PSR B.2 (see Figure B.1)

We validated the pipeline on two datasets: the DEAP dataset [67] and the WCS dataset (see Chapter 4). The methodology is described in Section B.3 and results are presented and discussed in Section B.4 and B.5.

### B.0.1 Notation

Physiological digital signals are a sampled version of a continuous variable  $s(t)$ . When the sampling rate, or sampling frequency, ( $f_{samp}$ ) is fixed, they

Symbol	Meaning	Symbol	Meaning
$\dot{D}$	Ideal driver function	$D$	Estimated driver function
$R_i$	Recovery portion of the peak $i$	$R_i^{dt}$	Detrended recovery portion of the peak $i$
$P_i$	4-seconds portion centered on the peak $i$	$D^{dp}$	<i>De-peaked</i> version of the estimated driver function
$\Phi$	Estimated phasic component	$T$	Estimated tonic component

Table B.1: Symbols used to indicate the main signals or signal components computed in the pipeline.

can be seen as a numerical series and indicated as  $s[n]$ . However, to ease the notation, when not explicitly required we omit the  $[n]$ ; in addition, we use capital letters to indicate the signals (or portions of signals); see Table B.1 for a summary of main symbols used to indicate the signals components computed in the pipelines.

## B.1 Loss function

Following the model proposed by Benedek and colleagues [13], the first step in the estimation of the triggering activity of the SNS is the computation of the driver function  $\dot{D}$ , which is performed by deconvolution.

In particular, we use the discrete version of the Bateman function as IRF:

$$B[n] = e^{\frac{-n}{N_1}} - e^{\frac{-n}{N_2}} \quad (\text{B.1})$$

where  $N_1 = \tau_1 f_{samp}$  and  $N_2 = \tau_2 f_{samp}$ . The two temporal constants  $\tau_1$  and  $\tau_2$  depend on physiological characteristics of the individuals and need to

be estimated.

In the ideal case,  $\dot{D}$  is a series of discrete delta impulses representing the triggering activity of the SNS (the PSR component) overlapped to the TSR component which derives from long-term physiological variations. In the real case the estimated driving function  $D$  is affected by errors due to: (a) existing noise in the acquired signal; (b) the deconvolution process which tends to amplify the noise; (c) the approximations due to the model.

For this reason, the optimization procedure aims at identifying the optimal parameters  $\tau_1$  and  $\tau_2$  to obtain a driving function similar to the ideal case. The optimization procedure is performed by minimization of a loss function which should represent the errors introduced in the deconvolution process. To identify an optimal loss function to be minimized we performed a test to investigate the effects of wrong estimates of  $\tau_1$  and  $\tau_2$  on the shape of the resulting deconvolved signal (see Figure B.2). We generated a surrogate EDA signal ( $f_{samp} = 8$  Hz) by convolution of a discrete delta impulse with a Bateman function ( $\tau_1 = 0.45s$ ,  $\tau_2 = 1.75s$ ). Then we deconvolved the surrogate signal using a set of Bateman functions with different values of  $\tau_1$  and  $\tau_2$ . We observed that wrong values of  $\tau_1$  and  $\tau_2$  alter the shape of the driving function after the peak instant: in particular we observe a longer recovery portion. In addition, the magnitude of the peak is also influenced. Based on these observations, we propose a novel loss function which estimates the energy of these alterations that is minimized by the optimization procedure.

The algorithm to compute the loss is regulated by the parameter  $\delta$  which is the expected amplitude of the minimum peak in the EDA signal. In literature this value varies between 0.01 and 0.05  $\mu S$ . In this study we set  $\delta = 0.02\mu S$ , however this value can be changed according to the experimental settings (e.g. device sensitivity, signal noise ratio). The parameter  $\delta$  is used as threshold value to discriminate between peaks associated to

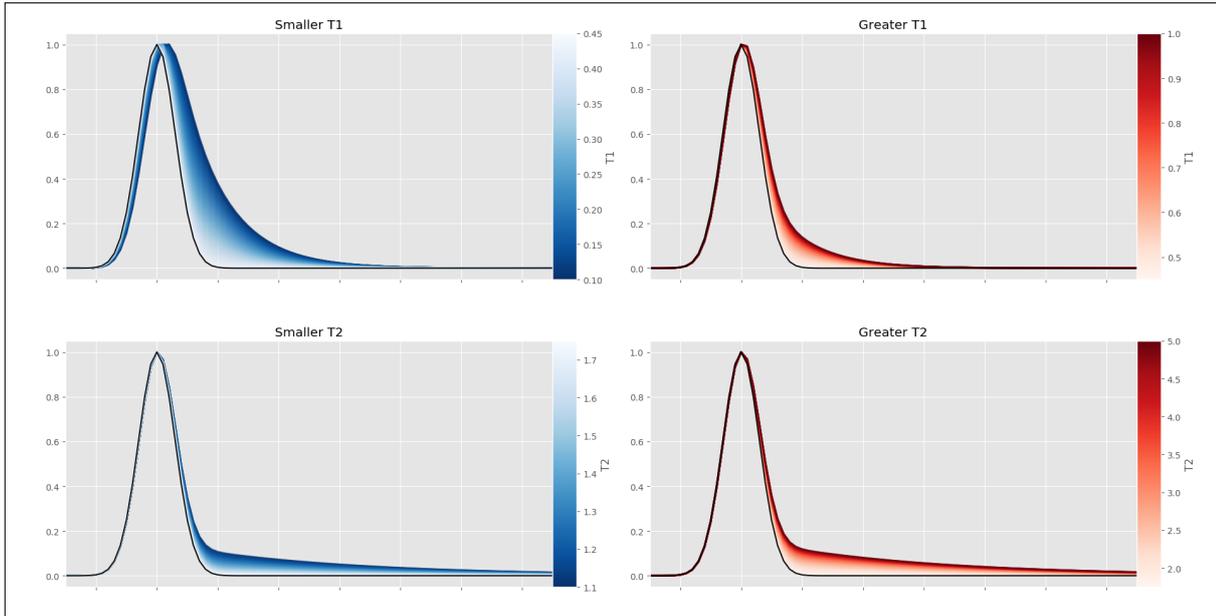


Figure B.2: Effects of wrong estimates of  $\tau_1$  and  $\tau_2$  on the recovery part of the resulting deconvolved signal. Top row: effects of smaller (left) and greater (right) values of  $\tau_1$ ; bottom row: effects of smaller (left) and greater (right) values of  $\tau_2$ . Signals have been normalized to have a maximum amplitude of 1.

SNS activity and peaks due to noise.

The loss function (see Figure B.1A) operates locally to compute the amount of scattered energy for each peak in  $D$ ; then the global loss is computed as average over all the peaks, as described in the following paragraphs.

It is worth noting that, differently from [13], the proposed loss function is computed directly on the estimated driver function  $\hat{D}$  and not on the PSR component. Thus, the optimization procedure depends only on the deconvolution process and not also on the method adopted to compute the TSR/PSR components.

### B.1.1 Stage 1: Identification of peaks

The first stage aims at identifying the peak instants  $p_i$  in  $D$ ; only peaks having amplitude greater than  $\delta$  are considered. The peak instants  $p_i$  are

used to segment the recovery portion of each peak ( $R_i$ ): the portion of  $D$  in the interval  $[p_i, p_i + W]$ , being  $W$  the number of samples corresponding to 10 seconds of signal. The recovery portion  $R_i$  will then be used in the following detrending procedure.

Note that it can occur that the distance between a peak instant  $p_i$  and the following  $p_{i+1}$  is less than 10s. We propose two different approaches to deal with this case. The first is to skip the peak  $i$ : this approach allows for a more precise detrending which would not be affected by the presence of overlapping peaks; however it would not be sustainable when analyzing signals with high density of peaks.

The second approach is to keep the peak  $i$ : this approach is prone to be less stable due to the effects of overlapping peaks, but it can be always applied. However, in the ideal case when optimal Bateman parameters are chosen, the effect of overlapping peaks is expected to be minimal.

### B.1.2 Stage 2: Detrending

The second stage aims at subtracting the linear support of  $R_i$  which represents the contribution of the TSR component. Two approaches are proposed, depending on the strategy adopted in the first stage to deal with overlapping peaks.

When overlapping peaks are skipped and only isolated peaks are considered the estimation of the linear support is based on the last 5 seconds portion of  $R_i$  that is used to fit a linear model. Figure B.3A exemplifies the procedure. The estimated linear support is then subtracted from  $R_i$  to obtain the detrended portion of the driving function corresponding to the peak  $i$ :  $R_i^{dt}$ .

This method is not applicable in case of overlapping peaks as a peak in last 5 seconds portion would affect the estimation of the linear model. In this case the linear model is estimated using the minima in the portion

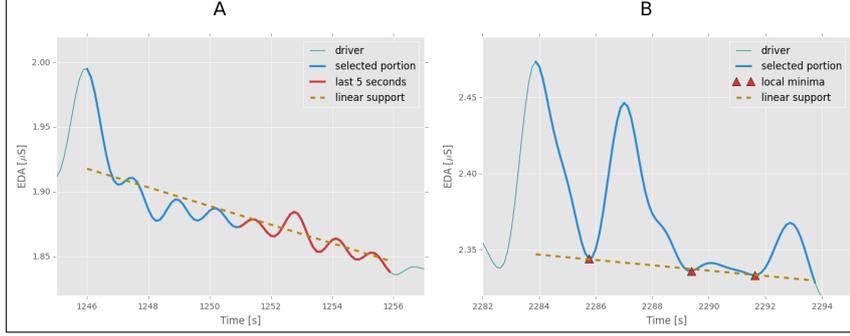


Figure B.3: Detrending in case of non overlapping (A) and overlapping (B) peaks.

(see Figure B.3B) and then subtracted from  $R_i$  to obtain  $R_i^{dt}$ .

### B.1.3 Stage 3: Computation of energy

To estimate the alteration in the driving function introduced by non-optimal Bateman parameters we compute the amount of energy after the peak in the detrended portion  $R_i^{dt}$ , normalized to account for the amplitude of the peak:

$$E_i = \frac{\sqrt{\sum_{k=1}^W (R_i^{dt}[k])^2}}{R_i^{dt}[1]} \quad (\text{B.2})$$

The global loss is then computed as the averaged energy over all the peaks:  $E = \frac{\sum_i E_i}{N}$ , where  $N$  is the number of peaks used to compute the loss.

## B.2 Estimation of PSR component

The method here proposed (see Figure B.1B) aims at improving the identification of inter peak intervals in  $D$  by analyzing its derivative:  $D'$ . The identification of inter peak intervals is a fundamental step to discriminate the PSR and TSR components. As for the loss function, the algorithm to estimate the PSR component is regulated by the parameter  $\delta$  with the

same meaning. A detailed explanation of the algorithm is given in the next paragraphs.

### B.2.1 Stage 1: Identification of candidate peaks

As in the first stage of the computation of the loss function, the algorithm operates first locally on each peak. The first step is the identification of peak instants in  $D$  associated to SNS activity. Then for each peak  $i$  the 4 seconds length portion centered on the peak ( $P_i$ ) is considered.

### B.2.2 Stage 2: Identification of onset and termination instants

The derivative  $P_i'$  is computed in order to find the onset instant  $n_i^o$  and the termination instant  $n_i^t$  of each peak  $i$ , as those instants immediately before and after  $p_i$  such as  $P_i'[k] = 0, k = n_i^o, n_i^t$ . Each detected peak in  $D$  is then substituted by a linear interpolation between each  $n_i^o$  and  $n_i^t$  to generate a *de-peaked* driver function  $D^{dp}$ .

### B.2.3 Stage 3: Interpolation

$D^{dp}$  is used to generate a 1 second-spaced grid to estimate the TSR component  $T$  by spline interpolation, as in [13]. Then the PSR component is computed as  $\Phi = D - T$ .

## B.3 Materials and Methods

We first tested the pipeline on the DEAP dataset [67]. The DEAP dataset contains physiological signals acquired with medical-grade device ( $f_{samp}=512$  Hz) collected from 32 subjects while watching 40 music videos (duration of stimuli: 60 seconds).

In addition, we tested the proposed pipeline on the stimuli portions of the WCS dataset (see Chapter 4) that has been appositely created in order to compare the performance on signals from wearable and clinical-grade devices.

### B.3.1 Metrics

As proposed in [13], we used the negativity index ( $\nu$ ) and the average width of the peaks ( $\psi$ ) to evaluate the performance of the pipeline. Both indexes are computed after optimization of the parameters  $\tau_1$  and  $\tau_2$  and estimation of the PSR component ( $\Phi$ ). The two metrics,  $\nu$  and  $\psi$ , aim at quantifying the distance of the computed  $\Phi$  from the ideal series of discrete delta impulses, therefore lower values indicate a better performance.

The negativity index  $\nu$  is computed as:

$$\nu = -\frac{\sum_k \Phi^\nu[k]}{N} \quad (\text{B.3})$$

where  $\Phi^\nu[k] = \Phi[k]$  if  $\Phi[k] < 0$ ,  $\Phi^\nu[k] = 0$  otherwise and  $N$  is the number of samples in the signal.

The average width index  $\psi$  is computed as:

$$\psi = \frac{\sum_k \Phi^\psi[k]}{n_p} \quad (\text{B.4})$$

where  $\Phi^\psi[k] = 1$  if  $\Phi[k] > \frac{\delta}{20}$ ,  $\Phi^\psi[k] = 0$  otherwise, and  $n_p$  is the number of detected peaks.

### B.3.2 Processing procedure

The EDA signal is first down sampled to 4 Hz then filtered to remove high frequency noise (infinite impulse response filter,  $f_{pass} = 0.8$  Hz,  $f_{stop} = 1.1$  Hz).

The optimization of the Bateman parameters to find the candidate  $\tau_1^c$  and  $\tau_2^c$  is performed using the Basin-Hopping stochastic algorithm (as implemented in the Python package `scipy` version 0.18.1), using  $\tau_1^c = 0.75$  and  $\tau_2^c = 2$  for initialization. Boundaries on the values of the parameters are set to:  $\tau_1 \in [0.1, 0.99]$ ,  $\tau_2 \in [1.5, 5]$ , the maximum number of iterations is set to 100 with option to terminate the optimization when the result is stable for 10 subsequent iterations.

The procedure is repeated on consecutive segments of the EDA signal extracted by windowing (length of the window: 180 s, shift between consecutive windows: 10 s) to generate multiple estimates of Bateman parameters for each subject. Then the average values  $\hat{\tau}_1$  and  $\hat{\tau}_2$  are used to generate the Bateman function to deconvolve the EDA signal and obtain the driving function. Finally, PSR component  $\Phi$  is estimated and metrics are computed.

In this work we compare three pipelines:

1. BEN: loss function to estimate Bateman parameters and PSR estimation algorithm as in [13];
2. P10: loss function based on the energy after the peak and rejection of peaks distant less than 10 seconds; PSR estimation algorithm based on the derivative of the driver function;
3. PKB: loss function based on the energy after the peak computed on all peaks and PSR estimation algorithm based on the derivative of the driver function.

All the computations have been performed in Python using `pyPhysio` (see Chapter 3) and `scipy` (version 0.18.1) packages.

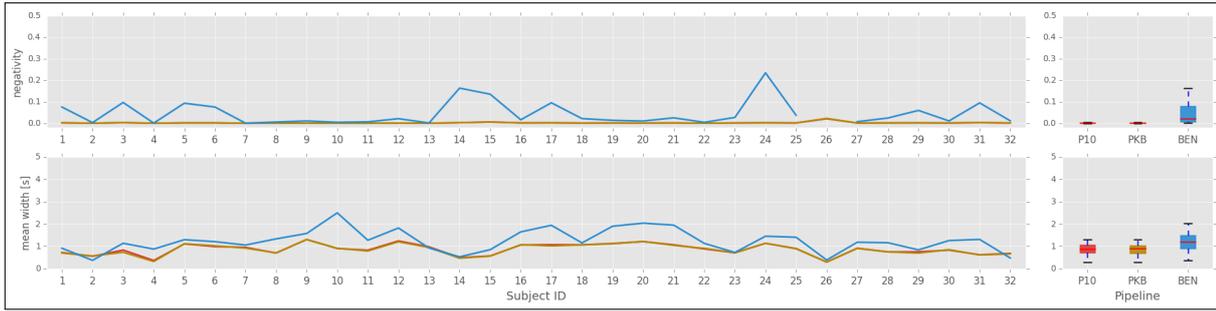


Figure B.4: Performances of the three pipeline on the DEAP dataset. By subject (left) and overall distribution (right).

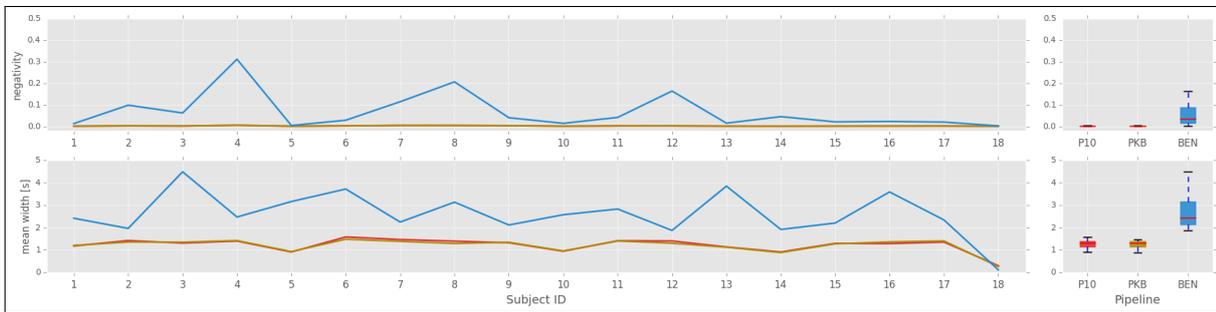


Figure B.5: Performances of the three pipelines on the WCS dataset. By subject (left) and overall distribution (right).

## B.4 Results

The two novel pipelines (P10 and PKB) provided a better performance than the state-of-art procedure (BEN) on both datasets, both in terms of negativity and mean width of peaks (see Figures B.4 and B.5).

Further, although values of estimated Bateman parameters differ (see Tables B.2 and B.3), P10 and PKB exhibit comparable performances. This suggests that for a good estimation of the PSR component the correct identification of the inter-peak intervals in the driver function is more important than the correct estimation of the Bateman parameters.

Regarding the analysis on the WCS dataset, we decided to focus on the EDA signals collected with the professional device (Infinity FlexComp), as EDA signals collected with Empatica E4 have a very poor quality and

Subject ID	$\tau_1$			$\tau_2$		
	P10	PKB	BEN	P10	PKB	BEN
1	0.56	0.67	0.59	2.19	2.56	2.16
2	0.7	0.85	0.76	2.3	2.54	2.02
3	0.47	0.78	0.69	2.1	2.56	2.08
4	0.82	0.83	0.76	2.17	2.85	2.14
5	0.69	0.71	0.78	2.23	2.77	2.13
6	0.63	0.73	0.51	2.05	2.59	2.26
7	0.72	0.86	0.75	2.28	3.29	2
8	0.78	0.72	0.82	2.43	3.15	2.18
9	0.78	0.82	0.8	2.07	3.96	2.2
10	0.79	0.68	0.76	2.35	3.35	2.3
11	0.8	0.8	0.77	2.38	3.46	2.15
12	0.63	0.81	0.79	2.36	2.94	1.98
13	0.78	0.87	0.81	2.18	2.84	2.24
14	0.73	0.79	0.74	2.1	2.94	2.4
15	0.51	0.68	0.52	2.02	2.57	2.45
16	0.52	0.75	0.64	2.09	2.58	2.35
17	0.46	0.65	0.7	2.01	2.24	2.04
18	0.67	0.67	0.68	2.04	3.3	2.19
19	0.63	0.61	0.79	2.17	2.98	2.35
20	0.83	0.75	0.8	2.09	2.98	2.31
21	0.62	0.61	0.7	2.29	2.92	2.39
22	0.77	0.85	0.67	2.42	3.28	2.45
23	0.68	0.58	0.57	2.04	2.34	2.11
24	0.46	0.69	0.46	2.07	2.76	2.43
25	0.53	0.79	0.39	2.09	2.5	2.05
26	0.7	0.52	0.75	2.04	2.07	2.06
27	0.59	0.71	0.77	2.22	2.51	2.06
28	0.6	0.74	0.59	1.93	2.72	2.45
29	0.34	0.59	0.59	2.06	2.3	2.03
30	0.71	0.68	0.63	1.94	2.55	2.46
31	0.58	0.77	0.55	1.88	2.62	2.3
32	0.61	0.84	0.62	2.45	2.49	2.09

Table B.2: Values of estimated Bateman parameters for the subjects in the DEAP dataset, according to different pipelines

Subject ID	$\tau_1$			$\tau_2$		
	P10	PKB	BEN	P10	PKB	BEN
1	0.71	0.62	0.62	2.04	2.77	2.36
2	0.57	0.68	0.73	2.08	2.5	2.28
3	0.64	0.8	0.62	2.43	2.55	2.12
4	0.73	0.75	0.66	2.08	2.32	2.11
5	0.69	0.52	0.73	2.19	2.57	2.42
6	0.69	0.73	0.8	2.29	2.66	2.05
7	0.5	0.82	0.72	2.11	2.38	2.15
8	0.36	0.74	0.53	1.95	2.14	2.42
9	0.75	0.71	0.64	1.99	2.8	2.59
10	0.74	0.73	0.68	1.98	2.78	2.21
11	0.86	0.72	0.87	2.1	2.54	2.53
12	0.44	0.73	0.63	1.89	2.77	2.34
13	0.61	0.69	0.71	2.24	3.16	1.98
14	0.57	0.65	0.66	1.89	2.22	2.48
15	0.55	0.73	0.63	2.24	3.02	2.24
16	0.76	0.7	0.73	2.4	2.64	2.33
17	0.81	0.77	0.77	2.6	3.09	2.19
18	0.77	0.8	0.82	2.43	2.77	2.39

Table B.3: Values of estimated Bateman parameters for the subjects in the WCS-Medical dataset, according to the three pipelines

it was not possible to perform the identification of the peaks. Results on the WCS dataset confirm that both P10 and PKB improve performances: both the negativity index and the mean width of the peaks are lower than those provided by the BEN pipeline.

## B.5 Discussion

According to the defined metrics, the P10 and PKB pipelines allowed better estimations of the PSR component, although it remains unclear whether the correct estimation of the Bateman parameters represents a fundamental step in the pipeline. In fact, the choice of the loss function adopted to optimize the Bateman parameters seemed not to significantly influence the

results; instead, it is probably the estimation of the PSR component of the driving function that provides the key improvement. However, the loss functions here proposed distinguish from the original one from Benedek and colleagues in the mathematical formulation and in the fact that they are computed on the driving function and not on the PSR component, thus being not dependent on the method adopted to isolate the TSR component.

An objective evaluation it is not possible due to the lack of a ground truth signal with known Bateman parameters. A better insight would be possible by performing additional simulated studies to be able to control the true value of the Bateman parameters.

Unfortunately, we were not able to investigate the performance of the proposed pipeline on EDA signal acquired with a wearable device as the signal obtained from the Empatica E4 had a very poor quality. Further investigation in this sense is therefore needed, possibly using a different device.



# Appendix C

## Wearable Devices

In this chapter we list a set of commercially available WDs that have been considered and tested in this work.

### C.1 Empatica E4

The Empatica E4 wristband (see Figure C.1A) embeds 4 sensors for: Blood Volume Pulse (BVP) for monitoring of cardiac activity [64 Hz], Electrodermal Activity (EDA) [4 Hz], Skin Temperature [4 Hz] and Acceleration [32 Hz]. Its main operational mode is online streaming, in which collected data are sent to a smartphone through Bluetooth Low Energy (BLE). It also features an offline acquisition mode in which data are saved on the local memory. In this case, to access data the device needs to be first connected to a computer where the Empatica Manager application takes care of uploading data to the Empatica server from where the user can then download them through the Empatica Dashboard web-interface. Autonomy of battery is about 24 hours. A Software Development Kit is provided to develop custom applications for data acquisition on Android. Otherwise, it is also possible to use the proprietary application, Empatica RT, which sends the data to the proprietary server from where the data can be downloaded. The device embeds a clock that automatically synchronizes



Figure C.1: Wearable devices tested in this work. A: Empatica E4 (picture from [www.empatica.com](http://www.empatica.com)); B: Mio Alpha (picture from [www.mioglobal.com](http://www.mioglobal.com)); C: ComfTech smart garments: t-shirt with acquisition unit; D: Interaxon Muse (picture from [www.choosemuse.com](http://www.choosemuse.com)); E: Emotiv Epoch (picture from [www.emotiv.com](http://www.emotiv.com)); F: EXEL Exls3 (picture from [www.exelmicroel.com](http://www.exelmicroel.com)).

with the computer clock once connected through the Empatica Manager application.

## C.2 Mio Alpha 2/Link

The Mio Alpha wristwatch (see Figure C.1B) embeds a sensor for BVP and a three axes accelerometer. Raw data are not accessible and are internally processed to return an indication of the Heart Rate in beats per minute. Local memory allows the storage of approximately 24 hours of data when the BLE streaming is not active; battery life is about 24 hours. Access to processed data and control of the device is available and custom applications for data acquisition can be developed. However, only HR reads are available, raw data can not be accessed.

### C.3 ComfTech smart garments

This category includes a group of sensorized garments (see Figure C.1C) produced by Comftech which embed textile electrodes connected to a small-size external removable electronic unit for acquisition of the electrocardiogram (ECG). The electronic unit also embeds a sensor for measuring acceleration. Sampling frequency can be set up to 200 Hz and data can only be streamed through Bluetooth as a local memory is not provided. Battery life has been empirically measured of about 5 hours. The design of garments is optimized and can be further customized according to the expected usage. In particular, the displacement of chest electrodes is minimized and the acquired signal is therefore less sensitive to moving artifacts. Access to raw data and control of the device is available through by the proprietary communication protocol, which is provided with the devices.

### C.4 Interaxon Muse

The Interaxon Muse headband (see Figure C.1D) acquires a 4 channels electroencephalogram (EEG) at 500 Hz sampling frequency. The commercial purpose is to improve meditation, however it can also be used for real-life monitoring of brain activity. The main feature is the adoption of dry electrodes, without any need of skin preparation. Communication is through BLE and no local memory is provided. An SDK is available to develop custom applications for data acquisition. However, this device has been only partially tested and an assessment of the signal quality and validity for scientific purposes is required.

## C.5 Emotiv Epoc/+

The Emotiv Epoc headset (see Figure C.1E) acquires a 14 channels EEG signal [256 Hz]. Unlike Interaxon Muse, Epoch+ employs wet electrodes, which limit usability but are expected to improve the quality of the signal. It communicates via BLE, battery life is about 12 hours and no local memory is provided. Raw data are accessible only through an SDK sold separately. The device has been only partially tested, stability of signal appeared to be discrete but accurate information about signal quality is not available.

## C.6 EXEL Exls3

The EXEL Exls3 device (see Figure C.1F) is an Inertial Measurement Unit (IMU) which embeds a tri-axial accelerometer, tri-axial gyroscope and tri-axial magnetometer for complete monitoring of activity and positioning. Sampling frequency can be set up to 300 Hz and data can be stored on the local memory (1 Gb) and/or streamed through Bluetooth. Battery autonomy is of about 2 hours. Access to raw data and control of the device is enabled by the proprietary communication protocol, which is provided with the devices.

# Appendix D

## pyPhysio tutorial

In this tutorial we first describe the usage of the two main classes in `pyPhysio`: `Signal` and `Algorithm`, then we provide two examples of signal processing pipelines for Heart Rate Variability (HRV) and Electrodermal Activity (EDA) analysis.

### D.1 Signals in `pyPhysio`

A signal is an ordered vector of timestamp-value pairs, where the timestamp is the instant at which the measured phenomenon had that value. In `pyPhysio` a signal is represented by the class `Signal` which extends the `numpy.ndarray` class. In this first section we will see the different types of signals that can be defined and their properties.

`pyPhysio` provides two subclasses of `Signal` to represent the physiological signals that can be collected during an experiment: `EvenlySignal` and `UnevenlySignal`. In the following code we import both, together with additional libraries used in these examples:

```
## import libraries
from __future__ import division
import numpy as np
import matplotlib.pyplot as plt
```

```
# import the Signal classes
from pyphysio import EvenlySignal, UnevenlySignal
```

Listing D.1: Import libraries and Signal classes

### D.1.1 EvenlySignal

When the signal is sampled with a fixed sampling frequency it is sufficient to know the timestamp at which the acquisition started and the sampling frequency to reconstruct the timestamp of each sample. This type of signal is represented by the class `EvenlySignal`.

Therefore, to create an instance of `EvenlySignal` these are the input attributes needed:

- `values` : values of the signal;
- `sampling_freq` : sampling frequency;
- `start_time` : temporal reference of the start of the signal. This is optional, if omitted it will be set to 0;
- `signal_nature` : identifier of the type of the signal. This information can be used to check the appropriateness of the algorithms applied to the signal. Now it is optional and if omitted it will be set to `'`.

In the following code we generate an `EvenlySignal` using as values some randomly generated numbers:

```
## create fake data
np.random.seed(4)
signal_values = np.random.uniform(0, 1, size = 1000)

## set the sampling frequency
fsamp = 100 # Hz

## set the starting time
```

```
tstart = 100 # s

## create the Evenly signal
s_fake = EvenlySignal(values = signal_values,
                      sampling_freq = fsamp,
                      start_time = tstart,
                      signal_nature = 'fake'
                      )

## plot
s_fake.plot()
```

Listing D.2: Create an EvenlySignal

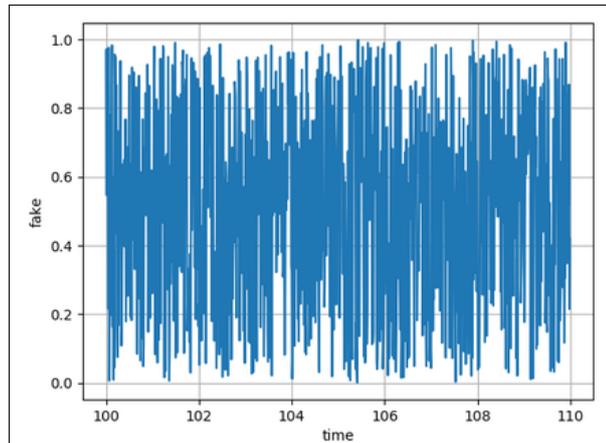


Figure D.1: Plot of the EvenlySignal created in Listing D.2

Class methods are provided to facilitate the management and processing of the signals:

- `get...`() type functions can be used to check signal attributes;
- `plot()` will plot the signal using the `matplotlib` library;
- `segment_time(t_start, t_stop)` and
- `segment_idx(idx_start, idx_stop)` can be used to extract a portion of the signal;

- `resample(fout)` can be used to change the sampling frequency.

In the following code we use the `get...()` type functions to check the attributes of the created `EvenlySignal`:

```
# check signal properties
print('Sampling freq.:{}'.format( s_fake.get_sampling_freq() ))
print('Start time:      {}'.format( s_fake.get_start_time() ))
print('End time:       {}'.format( s_fake.get_end_time() ))
print('Duration:      {}'.format( s_fake.get_duration() ))
print('Nature:        {}'.format( s_fake.get_signal_nature() ))
```

`pyPhysio` provides also some sample physiological signals that can be used for tests. In the following code we import the Electrocardiography (ECG) and EDA signals and create two `EvenlySignal` instances:

```
# import data from included examples
from pyphysio import TestData

ecg_data = TestData.ecg()
eda_data = TestData.eda()

# create two signals
fsamp = 2048
tstart_ecg = 15
tstart_eda = 5

ecg = EvenlySignal(values = ecg_data,
                   sampling_freq = fsamp,
                   signal_nature = 'ecg',
                   start_time = tstart_ecg)

eda = EvenlySignal(values = eda_data,
                   sampling_freq = fsamp,
                   signal_nature = 'eda',
                   start_time = tstart_eda)

# plot
ax1 = plt.subplot(211)
```

```
ecg.plot()
plt.subplot(212, sharex=ax1)
eda.plot()
```

Listing D.3: Import sample data provided in pyPhysio

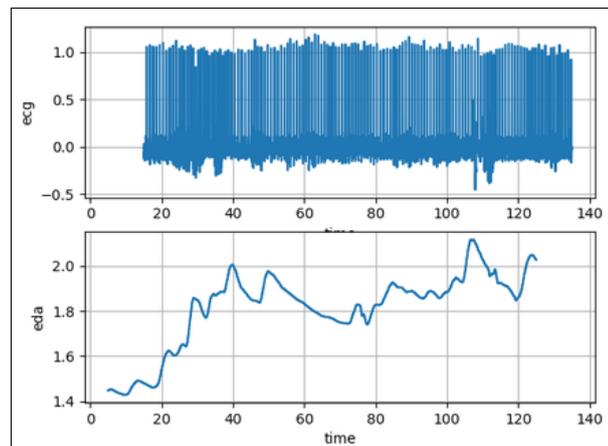


Figure D.2: Plot resulting from Listing D.3. Note the different starting time of the two signals.

### D.1.2 UnevenlySignal

Other types of signals, for instance triggers indicating occurrences of heartbeats or events, are series of samples which are not equally temporally spaced. Thus, the sampling frequency is not fixed and it is necessary to store the timestamp of each sample. This type of signals is represented by the class `UnevenlySignal`.

Therefore to create an instance of `UnevenlySignal` these are these additional input attributes are needed:

- `x_values` : information about the temporal position of each sample. Should be of the same size of `values`;
- `x_type` : 'instants' or 'indices' indicates what type of `x_values` has been used.

Two ways are allowed to define the temporal information when creating an `UnevenlySignal`:

1. by defining the indexes (`x_type='indices'`): `x_values` are indices of an array and the instants are automatically computed using the information from the `sampling_frequency` and the `start_time`;
2. by defining the instants (`x_type='instants'`): `x_values` are instants and the indices are automatically computed using the information from the `sampling_frequency` and the `start_time`.

As a general rule, the `start_time` is always associated to the index 0. In the following code we define two instances of an `UnevenlySignal` using both methods:

```
## create fake data
signal_values = np.arange(100)

## create fake indices
idx = np.arange(100)
idx[-1] = 125

## set the sampling frequency
fsamp = 10 # Hz

## set the starting time
tstart = 10 # s

## create an Unevenly signal defining the indices
x_values_idx = idx

s_fake_idx = UnevenlySignal(values = signal_values,
                             sampling_freq = fsamp,
                             signal_nature = 'fake',
                             start_time = tstart,
                             x_values = x_values_idx,
                             x_type = 'indices')
```

```
## create an Unevenly signal defining the indices
x_values_time = idx/fsamp + 10

## set the starting time
tstart = 0

s_fake_time = UnevenlySignal(values = signal_values,
                              sampling_freq = fsamp,
                              signal_nature = 'fake',
                              start_time = tstart,
                              x_values = x_values_time,
                              x_type = 'instants')
```

Listing D.4: Creation of UnevenlySignal instances

### D.1.3 Segmentation of signals

Two general class functions are provided to segment a signal:

1. `segment_time(t_start, t_stop)` is used to extract a portion of the signal between the instants `t_start` and `t_stop`;
2. `segment_idx(idx_start, idx_stop)` is used to extract a portion of the signal between the indices `idx_start` and `idx_stop`.

The output signal will inherit `sampling_freq` and `signal_nature` but the `start_time` will be set to `t_start` or to the instant corresponding to `idx_start` accordingly to the method used:

```
## segmentation of EvenlySignal
ecg_segment = ecg.segment_time(45, 54)
eda_segment = eda.segment_time(45, 54)

## plot
ax1 = plt.subplot(211)
ecg.plot()
ecg_segment.plot('r')
```

```
plt.subplot(212, sharex=ax1)
eda.plot()
eda_segment.plot('r')
```

Listing D.5: Segmentation of signals

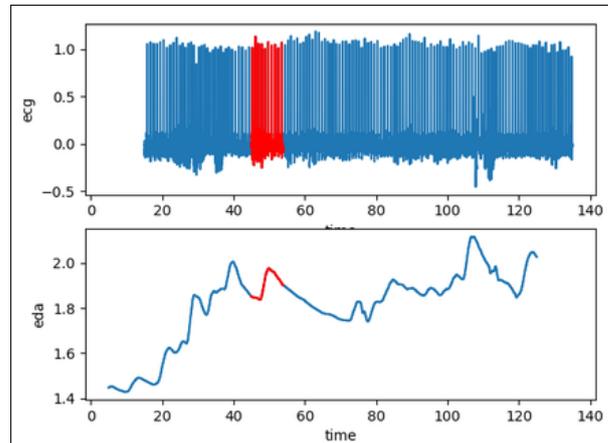


Figure D.3: Plot resulting from Listing D.5

## D.2 Algorithms in pyPhysio

A signal processing step is a computational function  $F$  that operates on an input signal to produce a result. It is characterized by a set of parameters  $\mathbf{p}$  that regulate its behavior (see Figure 3.2).

In `pyPhysio` each processing step is represented by an instance of a class derived from the generic class `Algorithm`. The type of function or algorithm is given by the Class name (e.g. `BeatFromECG` extracts the heartbeats from an ECG signal, `PeakDetection` detects the peaks in the input signal). The parameters of the function/algorithm are the attributes of the created instance.

Therefore, a processing step is defined by creating a new instance of the class, which is initialized with the given parameters:

```
processing_step = ph.BeatFromECG(parameters)
```

Listing D.6: Create a generic processing step

To execute the processing step we need to give as input an instance of the class `Signal`:

```
output = processing_step(input)
```

Listing D.7: Execute a generic processing step

Algorithms in `pyPhysio` are grouped in four classes:

1. **Filters** : deterministic algorithms that modify the values of the input signal without changing its nature;
2. **Estimators** : algorithms that aim at extracting information from the input signal which is given in output as a signal with a different nature;
3. **Indicators** : algorithms that operate on the signal to provide a scalar value (or metrics);
4. **Tools** : algorithms that can be useful for the signal processing and return as output one or more `numpy` arrays or scalars.

### D.2.1 Filters

`Filters` return a signal which has the same `signal_nature` of the input signal. The name `Filters` recalls the aim of these algorithms which is in general to increase the Signal/Noise ratio by filtering out the unwanted components in a signal (e.g.: high frequency noise). In the following snippet we create an Infinite Impulse Response (IIR) filter to cut the frequencies above 50 Hz:

```
import pyphysio.filters.Filters as flt

## create an IIRFilter
lowpass_50 = flt.IIRFilter(fp=50, fs=75, ftype='ellip')

## check parameters
print(lowpass_50)
```

```
## OR
lowpass_50.get()

## apply the IIRFilter
ecg_filtered = lowpass_50(ecg)

## plot
ecg.plot()
ecg_filtered.plot()
```

Listing D.8: Create and apply a Filter

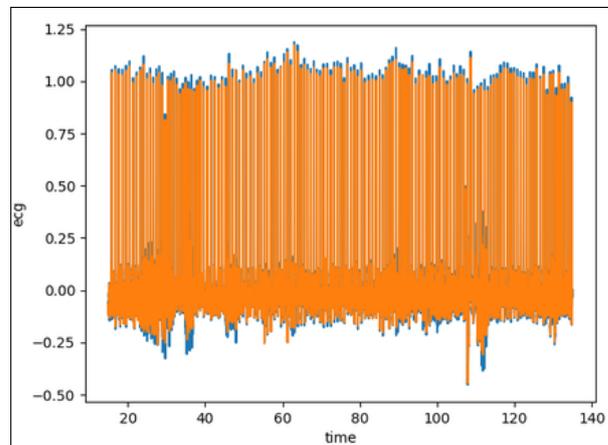


Figure D.4: Plot resulting from Listing D.8

## D.2.2 Estimators

Estimators are algorithms that aim at extracting the information of interest from the input signal, thus returning a new signal which has a different `signal_nature`.

The name `Estimators` recalls the fact that the information extraction depends on the value of the algorithm parameters which might not be known *a-priori*. Thus, the result should be considered as an estimate of the real content of information of the input signal.

```
import pyphysio.estimators.Estimators as est
```

```
## create a step to estimate the heart-beats
## position in the ECG signal
ibi_ecg = est.BeatFromECG()

## check parameters
ibi_ecg

## apply the estimator
ibi = ibi_ecg(ecg_filtered)

## plot
ax1 = plt.subplot(211)
ecg.plot()
plt.subplot(212, sharex=ax1)
ibi.plot()
```

Listing D.9: Create and apply an Eimator

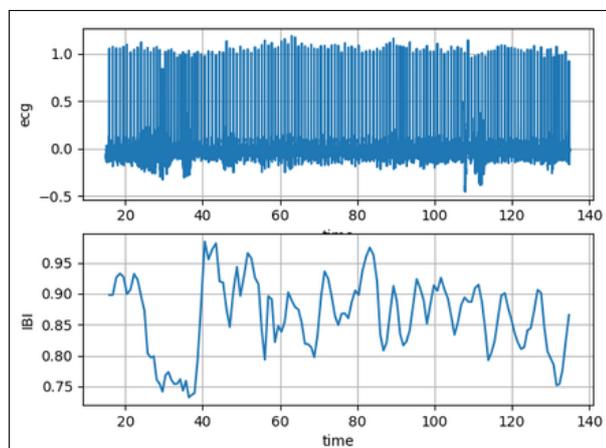


Figure D.5: Plot resulting from Listing D.9

### D.2.3 Indicators

Indicators are algorithms that extract a metrics (scalar value) from the input signal, for instance a statistic (e.g. average).

Three types of indicators are provided in `pyPhysio`:

1. Time domain indicators: comprising simple statistical indicators and other metrics that can be computed on the signal values;
2. Frequency domain indicators: metrics that are computed on the Power Spectrum Density (PSD) of the signal;
3. Non-linear indicators: complex indicators that are computed on the signal values (e.g. Entropy).

```
import pyphysio.indicators.TimeDomain as td_ind
import pyphysio.indicators.FrequencyDomain as fd_ind

## create the Root-mean-square of subsequent differences (RMSSD)
indicator
rmssd = td_ind.RMSSD()

## create the High Frequency band (HF) indicator
HF = fd_ind.PowerInBand(interp_freq=4, method = 'welch',
                        freq_min=0.15, freq_max=0.4)

## check parameters
print(rmssd)
print(HF)

## apply an Indicator
rmssd_ = rmssd(ibi)
HF_ = HF(ibi)

## print computed values
print(rmssd_)
print(HF_)
```

Listing D.10: Create and apply an Indicator

## D.2.4 Tools

This is a collection of useful algorithms that can be used for signal processing. These algorithms might return scalar values or `numpy` arrays. In the

following snippet we create a step to compute the Power Spectrum Density of the signal and plot the result:

```
import pyphysio.tools.Tools as t11

## create a step to extract the Power Spectrum Density
compute_psd = t11.PSD(method='welch', interp_freq = 4)

## check parameters
compute_psd

## apply a Tool
frequencies, power = compute_psd(ibi)

## plot
plt.plot(frequencies, power)
plt.show()
```

Listing D.11: Create and apply a Tools

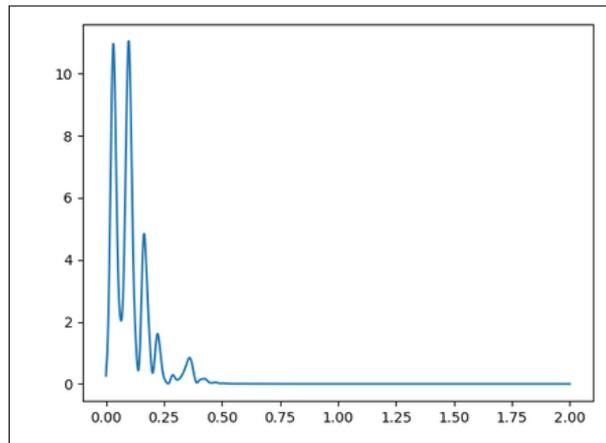


Figure D.6: Plot resulting from Listing D.11

### D.3 Pipelines in pyPhysio

In this section we describe two pipelines for the processing of ECG and EDA signals respectively.

We assume that the physiological signals are acquired during a multi-modal experiment composed of two sessions. The subject watches two images with different emotional content to elicit low and high arousal respectively. The objective of the experiment is to observe the physiological response associated to the different elicited emotional states. The experiment lasts 2 minutes, with 30 seconds of baseline at the beginning and 30 seconds of pause between the two images. In the following example we will artificially create an additional signal, called `label`, to store the information about the experimental timeline, i.e. which image is shown to the subject. The `label` signal will be used to assign a label to the indicators extracted during the windowing procedure to be able to compare the distributions of the values under the different stimuli.

We divide the pipelines into three separate steps, according to the categorization proposed in Chapter 3 (see Figure 3.1):

1. **Filtering and Preprocessing:** this step includes all the procedures aiming at increasing the signal/noise ratio, typically band-pass filtering, smoothing, removal of artifacts. The output of this step is a new version of the input signal with improved signal quality (reduced noise);
2. **Information Extraction:** this step aims at extracting the information of interest from the physiological signal. The output is a new signal containing only the information extracted and thus it has a different `signal_nature`;
3. **Physiological Indicators:** this step produces a list of scalar values able to describe the characteristics of the input signal. This step is usually performed on small segments of the input signals which are extracted using a sliding window on the whole length of the signal.

In the following examples we will also use the shortened syntax to apply a signal processing step:

```
## standard syntax: creation + execution

## creation of the processing step
filter_iir = ph.IIRFilter(fp=45, fs = 50, ftype='ellip')
## execution on the input signal
ecg_out = filter_iir(ecg)

## shortened syntax: creation(execution)
ecg_out = ph.IIRFilter(fp=45, fs = 50, ftype='ellip')(ecg)
```

Listing D.12: Explicit and shortened syntax for Algorithms

### D.3.1 ECG processing pipeline

In this pipeline we will process an ECG signal to estimate the Inter Beat Interval (IBI) signal and compute few time-domain HRV indicators from segments of the signal extracted by windowing (fixed size windows).

#### ECG Pipeline - Step 0: Import libraries and data

The starting point is importing `pyPhysio` and the other libraries required for the analysis, together with the provided sample data:

```
## import libraries
from __future__ import division
import numpy as np
import matplotlib.pyplot as plt

## import all pyphysio classes and methods
import pyphysio as ph

## import data and creating a signal
ecg_data = ph.TestData.ecg()

fsamp = 2048
ecg = ph.EvenlySignal(values = ecg_data, sampling_freq = fsamp,
                      signal_nature = 'ecg')
```

Listing D.13: ECG Pipeline - Step 0: Import libraries and create the signal

### ECG Pipeline - Step 1: Filtering and preprocessing

In this step we aim at increasing the signal/noise ratio of the signal by removing high-frequency noise. Then we normalize and over-sample the signal to improve the detection of heart beats:

```
## IIR filtering : remove high frequency noise
ecg = ph.IIRFilter(fp=45, fs=50, ftype='ellip')(ecg)

## normalization : normalize data
ecg = ph.Normalize(norm_method='standard')(ecg)

## resampling : increase the sampling frequency by cubic
interpolation
ecg = ecg.resample(fout=4096, kind='cubic')
fsamp = 4096
```

Listing D.14: ECG Pipeline - Step 1: Filtering and Preprocessing

### ECG Pipeline - Step 2: Information Extraction

The information we want to extract from the ECG signal is the position of the heartbeats and then the IBI signal:

```
ibi = ph.BeatFromECG()(ecg)

## check results so far
ax1 = plt.subplot(211)
ecg.plot()
plt.vlines(ibi.get_times(), np.min(ecg), np.max(ecg))
plt.subplot(212, sharex = ax1)
ibi.plot('o-')
plt.vlines(ibi.get_times(), np.min(ibi), np.max(ibi))
plt.show()
```

Listing D.15: ECG Pipeline - Step 2: Information Extraction

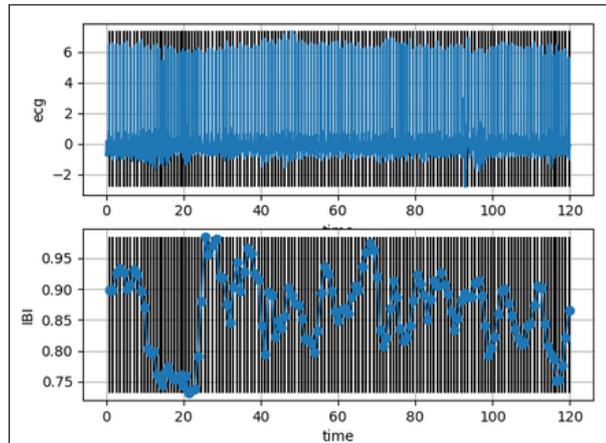


Figure D.7: Plot resulting from Listing D.15

### ECG Pipeline - Step 3: Physiological Indicators

Physiological indicators are computed over segments of the signal extracted by a windowing function, to track the time-variant characteristics of the Autonomic regulation.

The `label` signal is used to account for the experimental timeline and encodes the information about the different stimuli:

```
## create fake label
label = np.zeros(1200)
label[300:600] = 1 # watching image 1
label[900:1200] = 2 # watching image 2
label = ph.EvenlySignal(label, sampling_freq = 10,
                        signal_nature = 'label')

## plot
ax1 = plt.subplot(211)
ibi.plot('.-')
plt.subplot(212, sharex = ax1)
label.plot('.-')
plt.show()
```

Listing D.16: Creation of the `label` signal

In the following code we compute three time-domain HRV indicators

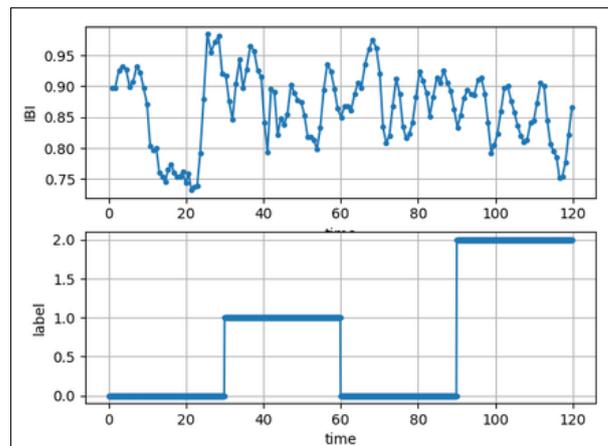


Figure D.8: Plot resulting from Listing D.16

using a windowing function which generates fixed-length and overlapping segments (length=10 s, step = 5 s). A Box and Whisker plot is generated to compare the distribution of the RRmean indicator under the two stimuli:

```
## define a list of indicators we want to compute
hrv_indicators = [ph.Mean(name='RRmean'),
                  ph.StDev(name='RRstd'),
                  ph.RMSSD(name='rmsSD')]

## fixed length windowing
fix_length = ph.FixedSegments(step = 5,
                              width = 10,
                              labels = label)

## computation of physiological indicators
hrv_ind, col_names = ph.fmap(fix_length, hrv_indicators, ibi)

## Box-Whisker plot
## extract the column with the label of each window
label_w = hrv_ind[:, np.where(col_names == 'label')[0]]

## extract the column with the RRmean values
## computed on each window
rrmean_w = hrv_ind[:, np.where(col_names == 'RRmean')[0]]
```

```

rrmean_image1 = rrmean_w[np.where(label_w==1)[0]]
rrmean_image2 = rrmean_w[np.where(label_w==2)[0]]

## create a box and whisker plot
## to compare the distribution of the RRmean indicator
plt.boxplot([rrmean_image1, rrmean_image2],
             labels=['image1', 'image2'])
plt.show()

```

Listing D.17: ECG Pipeline - Step 3: Physiological indicators

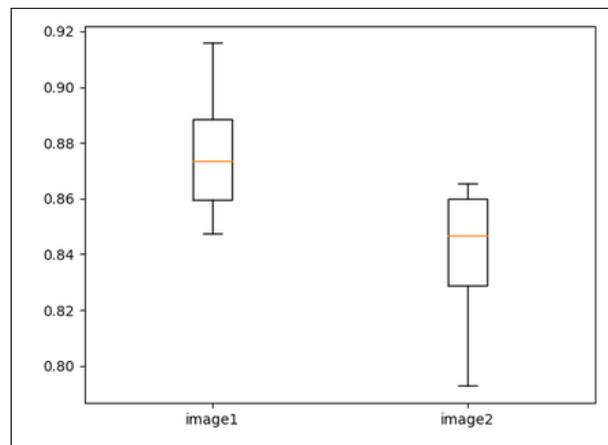


Figure D.9: Plot resulting from Listing D.17

Additional windowing methods are available: `LabelSegments` generates a segment for each continuous portion of the `label` signal having the same values; `CustomSegments` allows to manually define the start and stop instants of the segments:

```

## label-based windowing
label_based = ph.LabelSegments(labels = label)

## custom windowing
custom_based = ph.CustomSegments(begins = [0, 30, 60, 90],
                                 ends = [30, 60, 90,
                                          label.get_duration()])

```

Listing D.18: Additional windowing methods

### D.3.2 EDA processing pipeline

In this subsection we illustrate a pipeline for the analysis of an EDA signal:

```
eda_data = ph.TestData.eda()
eda = ph.EvenlySignal(values = eda_data,
                      sampling_freq = fsamp,
                      signal_nature = 'eda')
```

Listing D.19: EDA Pipeline - Step 0: Create the signal

In the preprocessing step we reduce the sampling frequency and filter out the high-frequency noise:

```
# decrease the sampling frequency by cubic interpolation
eda = eda.resample(fout=8, kind='cubic')

# remove high frequency noise
eda = ph.IIRFilter(fp=0.8, fs=1.1, ftype='ellip')(eda)
```

Listing D.20: EDA Pipeline - Step 1: Filtering and preprocessing

The information we want to extract from the EDA signal is the phasic component associated to the sympathetic activity:

```
## estimate the driver function
driver = ph.DriverEstim()(eda)

## compute the tonic and phasic components
phasic, tonic, tmp_ = ph.PhasicEstim(delta=0.02)(driver)

## check results so far
ax1 = plt.subplot(211)
eda.plot()
plt.subplot(212, sharex = ax1)
driver.plot()
phasic.plot()
tonic.plot()
plt.legend(['driver', 'phasic', 'tonic'])
plt.show()
```

Listing D.21: EDA Pipeline - Step 2: Information Extraction

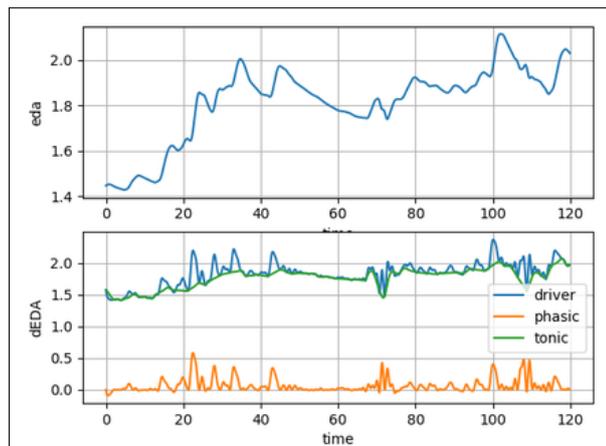


Figure D.10: Plot resulting from Listing D.21

Finally, we compute some time-domain indicators on the signal and on the characteristics of the peaks, then we visualize the results on a Box and Whisker plot. Note that we re-use the windowing function used to process the IBI signal to allow extracting the physiological indicators corresponding to the same segments:

```
## define a list of indicators we want to compute
indicators_eda = [ph.Mean(name='PhaMean'),
                  ph.StDev(name='StDev'),
                  ph.AUC(name='AUC'),
                  ph.PeaksMean(name='PksMean', delta=0.02),
                  ph.DurationMean(name='DurMean', delta=0.02)]

## compute the indicators on the phasic signal
pha_ind, col_names = ph.fmap(fix_length, indicators_eda, phasic)

## Box-Whisker plot
## extract column with the labels for each window
label_w = pha_ind[:, np.where(col_names == 'label')[0]]

## extract column with the PksMean values
## computed from each window
pksmean_w = pha_ind[:, np.where(col_names == 'PksMean')[0]]

pksmean_image1 = pksmean_w[np.where(label_w==1)[0]]
```

```
pksmean_image2 = pksmean_w[np.where(label_w==2)[0]]

## create a box and whisker plot
## to compute the distribution of the RRmean indicator
plt.boxplot([pksmean_image1, pksmean_image2],
            labels=['image1', 'image2'])
plt.show()
```

Listing D.22: EDA Pipeline - Step 3: Physiological indicators

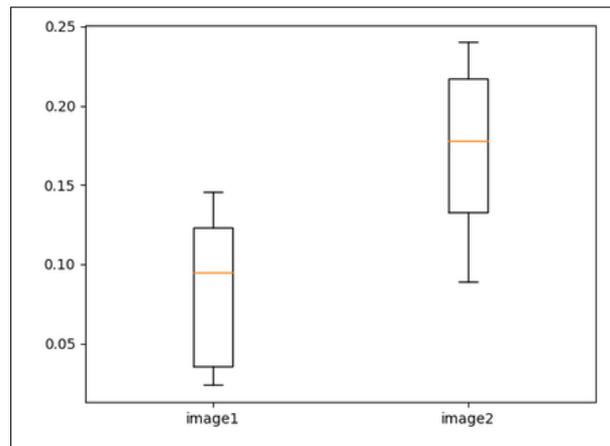


Figure D.11: Plot resulting from Listing D.22