

A Thesis Submitted for the Degree of PhD at the University of Warwick

Permanent WRAP URL:

<http://wrap.warwick.ac.uk/106454/>

Copyright and reuse:

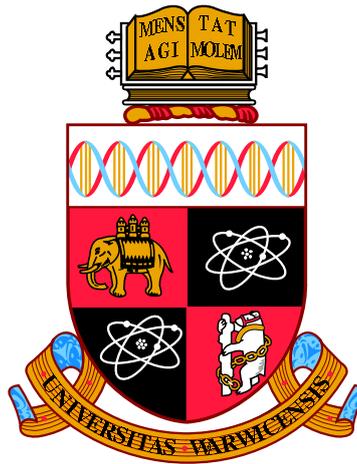
This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it.

Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk



**Monitoring Cardiovascular and Autonomic
Response in Real-life Settings**

by

Rossana Castaldo

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

School of Engineering

February 2018



Contents

List of Tables	vii
List of Figures	ix
Acknowledgments	xiii
Declarations	xiv
List of Publications	xvi
Journal papers	xvi
Peer reviewed full conference papers	xvi
Book chapter	xviii
Abstract	xix
List of Abbreviations	xx
Chapter 1 Introduction to the Research	1
1.1 Chapter overview	1
1.2 Introduction to the research topic	1
1.3 The rationale of the research	2
1.4 Research questions, aim and objectives	3
1.5 Research methodology	6
1.6 The structure of the thesis	10
Chapter 2 Background	13
2.1 Chapter overview	13
2.2 Cardiovascular and Autonomic Nervous Systems	13
2.2.1 Heart Rate Variability	16
2.2.1.1 HRV Analysis	18

2.2.1.2	Factors influencing heart rate and its variability . . .	30
2.3	Theoretical limits of biomedical signal processing and machine learning in real-life settings	32
2.3.1	Biomedical signal processing in real-life settings	33
2.3.2	Data-mining and machine learning analysis in real-life settings	35
2.4	Data mining	39
2.4.1	Signal pre-processing and features extraction	42
2.4.2	Feature selection and data-driven machine learning techniques	47
2.4.2.1	Common machine learning methods	51
2.5	Conclusions	54

Chapter 3 Literature Review on Acute Mental Stress and Falls in

	Later-life	56
3.1	Chapter overview	56
3.2	Literature review on acute mental stress detection via HRV	57
3.2.1	Stress definition	57
3.2.2	Systematic literature review with meta-analysis	58
3.2.2.1	Methods and materials	58
3.2.2.2	Search strategy	58
3.2.2.3	Inclusion and exclusion criteria for paper selection .	59
3.2.2.4	Paper short-listing, data extraction and outcomes of interest	59
3.2.2.5	HRV features	60
3.2.2.6	Statistical analysis and software tools	61
3.2.2.7	Matlab tool for meta-analysis	63
3.2.2.8	Results	67
3.2.2.9	Characteristics of the included studies	68
3.2.2.10	Trends of the HRV features	70
3.2.2.11	Pooled pivot values of HRV features	73
3.2.2.12	Models to detect mental stress via short HRV features	78
3.2.2.13	Discussion	78
3.2.2.14	Conclusion and recommendations	80
3.2.3	Literature review on methods to assess ultra-short HRV features	81
3.2.3.1	Methods and materials	82
3.2.3.2	Results and discussion	83
3.2.3.3	Conclusion	91
3.3	Literature review on accidental falls in later-life	91

3.3.1	Fall definition	92
3.3.2	Risk factors	93
3.3.3	Fall prevention	95
3.3.3.1	Prevention tools	95
3.3.4	Fall prediction tools	96
3.3.5	Monitoring technologies for fall detection and prediction . . .	97
3.3.5.1	Non-wearable sensors	98
3.3.5.2	Wearable sensors	99
3.3.6	HRV and fall prediction	100
3.3.7	Conclusion and limitations	102
3.4	Conclusions	103

Chapter 4 Development of Methods and Tools to Monitor Cardiovascular and Autonomic Response in Real-life Settings 105

4.1	Chapter overview	105
4.2	Theoretical approaches to biomedical signal processing and machine learning in real-life settings	105
4.2.1	How to determine surrogates of biomedical signals	106
4.2.1.1	Framework to assess the validity of ultra-short HRV features in a control condition	109
4.2.1.2	Framework to assess the validity of ultra-short HRV features under two conditions	111
4.2.1.3	Matlab tool to investigate biomedical surrogates	115
4.2.2	Data-driven machine learning techniques for biomedical signals in real-life settings	121
4.2.2.1	How to cope with small balanced datasets	121
4.2.2.2	Matlab tool to develop an automatic classifier using small balanced datasets	127
4.2.2.3	How to cope with unbalanced datasets	132
4.2.2.4	Matlab tool to develop automatic classifier using unbalanced datasets	136
4.3	Conclusions and limitations	141

Chapter 5 Cardiovascular and Autonomic Response to Mental Stress 142

5.1	Chapter overview	142
5.2	Detection of mental stress in real life	144
5.2.1	Dataset	144
5.2.2	Hardware and software	145

5.2.3	Data analysis	146
5.2.4	HRV analysis	147
5.2.5	Statistical analysis	153
5.2.6	Multi-scale HRV comparison: short VS ultra-short	153
5.2.6.1	Non-parametric statistical significance and trend analyses	154
5.2.6.2	Correlation analysis and Bland-Altman plots	154
5.2.6.3	Feature subset selection	155
5.2.7	Data-driven machine learning	156
5.2.7.1	HRV feature selection	157
5.2.7.2	Machine learning methods	157
5.2.7.3	Training, validation and testing	158
5.2.8	Results	159
5.2.8.1	Statistical analysis	159
5.2.8.2	Multi-scale HRV comparison: short VS ultra-short	163
5.2.8.3	Classification and performance measurements	165
5.2.9	Discussion	168
5.2.10	Conclusion and applications	172
5.3	Detection of mental stress in laboratory settings	173
5.3.1	Experiment designs	173
5.3.1.1	Study population	174
5.3.1.2	Stroop Word Colour Test (SCWT) and Video Game Challenge (VGC) implementation	177
5.3.1.3	Ethical approvals	179
5.3.1.4	Study protocol	180
5.3.1.5	SCWT protocol	180
5.3.1.6	VGC protocol	181
5.3.2	Hardware and software	183
5.3.3	Data analysis	184
5.3.4	HRV analysis	185
5.3.5	Statistical analysis	188
5.3.5.1	Comparison real VS in-lab stressors: exploratory analysis	189
5.3.6	Data-driven machine learning	190
5.3.7	Results	191
5.3.7.1	Statistical analysis	191

5.3.7.2	Comparison real VS in-lab stressors: exploratory results	194
5.3.7.3	Classification and performance measurements	199
5.3.8	Discussion	202
5.3.9	Conclusion	204
5.4	Conclusions and limitations	205
Chapter 6 Cardiovascular and Autonomic Response to Falls in Later-life		207
6.1	Chapter overview	207
6.2	Dataset	209
6.3	Hardware and software	210
6.4	Data analysis	211
6.5	Short term HRV analysis	212
6.6	Statistical analysis	216
6.7	Data-driven machine learning	216
6.7.1	HRV feature selection	217
6.7.2	Machine learning methods	218
6.7.3	Training, validation and testing	218
6.7.3.1	Final model generation	219
6.8	Results	220
6.8.1	Statistical analysis	220
6.8.2	Classification and performance measurements	221
6.9	Discussion	225
6.10	Conclusions, applications and limitations	226
Chapter 7 Conclusions and Future Work		228
7.1	Chapter overview	228
7.2	Research aim	228
7.3	Contribution to the body of knowledge	229
7.4	Research questions and answers	231
7.5	Research objectives: summary of findings and conclusions	232
7.6	Limitations and future work	237
References		240
Appendices		262

Appendix A Matlab Tools	263
A.1 Matlab tool for meta-analysis	263
A.2 Matlab tool to identify biomedical surrogates	269
A.3 Matlab tool to develop an automatic classifier using small balanced datasets	308
A.4 Matlab tool to develop an automatic classifier using unbalanced datasets	328
 Appendix B Supplementary Materials	 352
B.1 Bland-Altman plots	352
B.2 Questionnaire and ethical approvals	359

List of Tables

1.1	Summary of the explored case studies, objectives and deliverables. . .	5
1.2	Signal analysis and acquisitions.	6
2.1	HRV features in the time domain.	20
2.2	HRV features in the frequency domain.	22
2.3	Non-linear HRV features.	30
2.4	The confusion matrix for a binary classification problem.	50
2.5	Measures for binary classification using the nation of Table 2.4. . . .	50
3.1	Characteristics of studies included in the review.	68
3.2	Description of study designs included in the review.	70
3.3	Extracted time domain HRV features.	71
3.4	Extracted frequency domain HRV features.	72
3.5	Extracted non-linear HRV features.	73
3.6	Pooled HRV features.	74
3.7	Characteristics of studies investigating ultra-short HRV.	88
3.7a	Characteristics of studies investigating ultra-short HRV (cont). . . .	89
3.8	Characteristics of the models aiming to detect stress via ultra-short HRV features.	91
3.9	Fall prevention, detection and prediction.	93
3.10	Mobility tests proposed in literature for predicting falls.	97
3.11	HRV features from 24h in fallers and non-fallers [224].	101
3.12	Best performance of the adopted classification methods using HRV features from 24h in fallers and non-fallers [64].	102
5.1	Heart Rate Variability (HRV) features.	153
5.2	HRV features in rest and stress from 5 min NN data series. Academic Examination (AE) experiment.	159

5.3	HRV features in rest and stress from 3 min NN data series. AE experiment.	160
5.4	HRV features in rest and stress from 2 min NN data series. AE experiment.	161
5.5	HRV features in rest and stress from 1 min NN data series. AE experiment.	162
5.6	HRV features in rest and stress from 30 sec NN data series. AE experiment.	162
5.7	HRV features' trends.	163
5.8	Correlation analysis of ultra-short HRV VS short HRV features. . . .	165
5.9	Correlation among HRV features in Folder 1.	166
5.10	Model performance measurements estimated on the test set (Folder 2) for 5 min excerpts.	166
5.11	Model performance measurements on different time-scale excerpts. .	167
5.12	HRV features in rest and stress from 5 min NN data series. Stroop Colour Word Test (SCWT) experiment.	192
5.13	HRV features in rest and stress from 1 min NN data series. SCWT experiment.	193
5.14	HRV features in rest and stress from 5 min NN data series. Video Game Challenge (VGC) experiment.	194
5.15	HRV features in rest and stress from 1 min NN data series. VGC experiment.	194
5.16	HRV features' trends during real and in-lab stressors.	196
5.17	Comparison between real and in-lab stressors.	197
5.18	Correlation among HRV features in Folder 1.	200
5.19	Combinations of relevant and non-redundant HRV features.	200
5.20	Model performance measurements estimated on test set (Folder 3) on 1 min excerpts.	201
6.1	HRV features.	215
6.2	Patient baseline characteristics.	220
6.3	HRV features in non-fallers and fallers.	221
6.4	Correlation among HRV features in Folder 1.	223
6.5	Performance measurements (Mean \pm SD) estimated on the test set (Folder 3).	224
7.1	Summary of the main results.	233

List of Figures

1.1	Research process.	9
1.2	The thesis at a glance.	12
2.1	Sympathetic and Parasympathetic branches [26].	15
2.2	Raw ECG trace.	16
2.3	Diagram of the Cardiovascular Control System.	17
2.4	Electrophysiology of the heart [33].	18
2.5	Power Spectrum Density (PSD) estimation using an Auto-Regressive (AR) method.	22
2.6	Example of Poincaré plot features.	24
2.7	DFA example.	28
2.8	RP example.	29
2.9	Data analysis for wearable sensor data. Adapted from [69].	33
2.10	Artificial intelligence, machine learning and deep learning.	35
2.11	Difference between traditional machine learning and deep learning.	36
2.12	Theoretical limitations of applied biomedical signal processing and data-driven machine learning in real-life settings.	39
2.13	Data mining tasks. Adapted from [69].	41
2.14	Different types of machine learning methods and categories of algorithms.	42
2.15	Data mining main steps.	42
2.16	Flowchart summarising the individual steps when pre-processing ECG signals to obtain data for HRV analysis.	44
2.17	An example of NN interval time series.	47
2.18	An example of ROC curve from one of the studies.	51
3.1	Pseudocode for the meta-analysis tool.	63
3.2	An example of input matrix for the meta-analysis Matlab tool.	64
3.3	An example of output for the meta-analysis Matlab tool.	64

3.4	UML sequence graph of the Matlab tool for meta-analysis.	66
3.5	Flowchart of literature search: titles, abstract and full-papers included/excluded.	67
3.6	Forest plots of the pooled HRV features.	75
3.6a	Forest plots of the pooled HRV features (cont.).	76
3.6b	Forest plots of the pooled HRV features (cont.).	77
3.7	Flowchart of the literature review. An overview of methods employed in the shortlisted 29 papers to assess the validity of ultra-short HRV features.	85
3.8	Risk factors for falls.	94
4.1	The algorithm to select surrogates in one condition.	110
4.2	Methodological framework to assess the validity of ultra-short HRV features as surrogates of 5 min HRV ones in a control condition. . .	111
4.3	Methodological framework to assess the validity of ultra-short HRV features as surrogates of 5 min HRV ones to detect an adverse event.	113
4.4	Pseudocode to assess the validity of surrogates.	114
4.5	An example of input matrix. Observation are arranged in rows and HRV features in columns.	115
4.6	An overview of the developed tool to investigate surrogates.	115
4.7	UML sequence graph of the Matlab tool for surrogate analysis for one condition.	119
4.8	UML sequence graph of the Matlab tool for surrogate analysis for two conditions.	120
4.9	Splitting of the dataset into two folders.	123
4.10	Framework for feature selection.	124
4.11	Model training, validation and testing for small balanced datasets. . .	126
4.12	An overview of the developed tool to develop classifiers using small datasets.	127
4.13	Pseudocode of the tool for small datasets.	128
4.14	UML sequence graph of the Matlab tool for small datasets.	131
4.15	UML sequence graph of the Matlab tool for the feature selection process.	132
4.16	Splitting of the dataset into three folders.	134
4.17	Model training, validation and testing for unbalanced datasets.	135
4.18	An overview of the developed tool to develop classifiers using unbalanced datasets.	136
4.19	Pseudocode for the tool for unbalanced datasets.	137

4.20	UML sequence graph of the Matlab tool for unbalanced datasets. . .	140
5.1	Workflow for Case Study 1.	143
5.2	Data analysis flow for real-life stress.	147
5.3	Raw NN series for one subject during the AE rest and stress sessions over different time scales (i.e., 5 min, 3 min, 2 min, 1 min and 30 sec.)	149
5.4	HRV processing workflow for real-life stress.	150
5.5	Excerpt extraction.	151
5.6	Segmentation process for real-life stress.	151
5.7	Methodological workflow for the identification of the good surrogates.	156
5.8	ROC curves of the K-Nearest Neighbour Classifier (IBK) final model for different time scale excerpts.	167
5.9	The minimum sample for predictive modelling.	176
5.10	SWCT example slides	178
5.11	VGC interface.	179
5.12	SCWT protocol.	181
5.13	VGC protocol.	182
5.14	Zephyr BioPatch (on the left-hand side) and Zephyr BioPatch side strap (on the right-hand side).	183
5.15	The NeXus-10 (MindMedia).	184
5.16	Data analysis flow for in-lab stress.	185
5.17	Raw NN series for one subject during the SCWT rest and stress sessions over 5 min and 1 min excerpts.	186
5.18	Raw NN series for one subject during the VGC rest and stress sessions over 5 min and 1 min excerpts.	186
5.19	The HRV processing workflow for in-lab stress.	187
5.20	Segmentation process for in-lab stress.	188
5.21	Conceptual difference between the within- and between-group tests .	190
5.22	Data-driven machine learning workflow.	191
5.23	Profile plots of median and Standard Error (SEM) for the Rest Con- dition (RC) and the Stress Condition (SC) using different stressors. AE: Academic Examination; SCWT: Stroop Colour Word Test; VGC: Video Game Challenge; - :dimensionless.	198
5.23a	Profile plots of median and Standard Error (SEM) for the Rest Con- dition (RC) and the Stress Condition (SC) using different stressors. AE: Academic Examination; SCWT: Stroop Colour Word Test; VGC: Video Game Challenge; - :dimensionless (cont.).	199

5.24	IBK performance against HRV features combinations.	201
5.25	ROC curve for the IBK final model.	202
6.1	Workflow for Case Study 2.	209
6.2	Data analysis flow.	212
6.3	HRV processing workflow	213
6.4	Raw NN series for one faller and non-faller during baseline session for over an hour.	214
6.5	Splitting of the dataset into three folders.	217
6.6	Model training, validation and testing.	219
6.7	ROC curve of the Multinomial Naïve Bayes' final model.	225
7.1	ROC curve comparison for Case Study 1.	230
7.2	ROC curves comparison for Case Study 2.	231
7.3	Roadmap for future work to improve mental stress detection.	238
7.4	Roadmap for future work to improve fall prediction in later-life	239
B.1	Bland-Altman plots for MeanNN across time scales during rest and stress conditions.	353
B.2	Bland-Altman plots for StdNN across time scales during rest and stress conditions.	354
B.3	Bland-Altman plots for MeanHR across time scales during rest and stress conditions.	355
B.4	Bland-Altman plots for StdHR across time scales during rest and stress conditions.	356
B.5	Bland-Altman plots for HF across time scales during rest and stress conditions.	357
B.6	Bland-Altman plots for SD2 across time scales during rest and stress conditions.	358

Acknowledgments

In the years spent working on this Ph.D. thesis, I am glad I met wonderful people around me.

I would like to thank my university supervisors Dr Leandro Pecchia and Prof. Christopher James for giving me the opportunity to embark on this Ph.D., for generously sharing their ideas and support and for all opportunities to learn and develop at courses, conferences and work-shops.

My family to be always by my side and support my decisions even though they brought me far away from them. Thanks for making the world smaller by always being there: on Skype and visiting me every time you could. I love you.

My fiancè and best friend Peppe for staying by my side when I needed the most, for his optimism, support and love and for repeatedly teaching me to be more confident in my ideas and remember what really matters in life.

My bestie Principia to never make me feel alone and constantly remind me to take care of myself.

My colleagues and friends in the Biomedical lab for the moral support and laughs. Rucha, Flick, Yan, Vindy and Abi for motivating me when I was feeling down. My friend and colleague Luis for the encouragement and help during these years.

Declarations

I declare that the work presented in this thesis is my own except where stated otherwise, and was carried out entirely at the University of Warwick, during the period from September 2014 to February 2018, under the valuable supervision of Dr Leandro Pecchia and significant help from Prof. Christopher James. The research reported here has not been submitted, either wholly or in part, in this or any other academic institution for admission to a higher degree.

This thesis contains fewer than 64000 words excluding appendices, bibliography, footnotes, and tables and has fewer than 45 and 78 tables and figures respectively.

The work presented was carried out by the author except in the cases outlined below:

- the data reported in Chapter 5, section 5.2, were acquired prior to the work performed in this thesis and not by the author. This work was focused on data analysis and not on data acquisition;
- the data reported in Chapter 6 were acquired prior to the work performed in this thesis and not by the author. This work was focused on data analysis and not on data acquisition.

Some parts of the work reported and other works not reported in this thesis have been published. Further parts of this work will be submitted for publication in due course. In particular, parts of this thesis have been published by the author:

- Chapter 3 presents the materials from a published review [1], preliminary results presented in a conference paper [2] and a manuscript under revision [3].

- Chapter 4 presents an extended version of the methods presented in a published manuscript [4] and a manuscript under revision [3].
- Chapter 5 presents the materials originating from a manuscript under revision [5], preliminary results presented in different conference papers [6–9].
- Chapter 6 presents the materials from a published manuscript [4] and two conference papers [10, 11].

List of Publications

Journal papers

1. Castaldo R., Melillo P., Izzo R., De Luca N. and Pecchia L.,(2017) Fall Prediction in Hypertensive Patients via Short-Term HRV Analysis. *IEEE Journal of Biomedical and Health Informatics*, 21, 2, 399-406.
2. Castaldo, R., Melillo, P., Bracale, U., Caserta, M., Triassi, M., and Pecchia, L. (2015). Acute mental stress assessment via short term HRV analysis in healthy adults: A systematic review with meta-analysis. *Biomedical Signal Processing and Control*, 18, 370-377.
3. Castaldo R., Montesinos L., Melillo P., James C. and Pecchia L.,(2017). Ultra-short term HRV Features as Surrogate of Short term HRV. A Case Study on Mental Stress Detection in Real Life. *BMC Medical Informatics and Decision Making* (accepted).
4. Pecchia L., Castaldo R., Montesinos L.,and Melillo P., (2017). Are ultra-short Heart Rate Variability features good surrogates of short term ones? Literature review and method recommendations. *Healthcare Technology Letters* (under review).
5. Montesinos, L., Castaldo, R., and Pecchia, L. (2018). Wearable Inertial Sensors for Fall Risk Assessment and Prediction in Older Adults: A Systematic Review and Meta-Analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* (accepted).

Peer reviewed full conference papers

1. Castaldo, R., Montesinos, L., Wan, T. S., Serban, A., Massaro, S., and Pecchia, L. (2017). Heart Rate Variability Analysis and Performance during a Repeated

- Mental Workload Task. In *EMBEC and NBC 2017* (pp. 69-72). Springer, Singapore.
2. Castaldo, R., Montesinos, L., Melillo, P., Massaro, S., and Pecchia, L. (2017). To What Extent Can We Shorten HRV Analysis in Wearable Sensing? A Case Study on Mental Stress Detection. In *EMBEC and NBC 2017* (pp. 643-646). Springer, Singapore.
 3. Castaldo R., Xu, W., Melillo, P., Pecchia, L., Santamaria, L., and James, C. (2016). Detection of mental stress due to oral academic examination via ultra-short-term HRV analysis. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the* (pp. 3805-3808). IEEE.
 4. Castaldo, R., and Pecchia, L. (2016). Preliminary Results from a Proof of Concept Study for Fall Detection via ECG Morphology. In *XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016* (pp. 205-208). Springer International Publishing.
 5. Castaldo, R., Melillo, P., and Pecchia, L. (2015). Acute Mental Stress Detection via Ultra-short term HRV Analysis. In *World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada* (pp. 1068-1071). Springer International Publishing.
 6. Castaldo R., Melillo, P., and Pecchia, L. (2015). Acute mental stress assessment via short term HRV analysis in healthy adults: a systematic review. In *6th European Conference of the International Federation for Medical and Biological Engineering* (pp. 1-4). Springer International Publishing.
 7. Montesinos, L., Castaldo, R., Piaggio, D., and Pecchia, L. (2017). Day-to-day variation in sleep quality and static balance: results from an exploratory study. In *EMBEC and NBC 2017* (pp. 611-614). Springer, Singapore.
 8. Meccariello, P., Castaldo, R., Montesinos, L., Guglielmelli, E., and Pecchia, L. (2016). A Matlab Tool to Support Systematic Literature Review with Meta-Analysis. In *XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016* (pp. 994-996). Springer International Publishing.
 9. Melillo, P., Castaldo, R., Sannino, G., Orrico, A., De Pietro, G., and Pecchia, L. (2015). Wearable technology and ECG processing for fall risk assessment,

prevention and detection. In Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE (pp. 7740-7743). IEEE.

10. Castaldo, R., Montesinos, L., and Pecchia, L. (2018). Ultra-short Entropy for mental stress detection. In World Congress on Medical Physics and Biomedical Engineering, June 3-8, 2018, Prague, Czech Republic (accepted).
11. Montesinos, L., Castaldo, R., and Pecchia, L. (2018). Selection of entropy-measure parameters for force plate-based human balance evaluation. In World Congress on Medical Physics and Biomedical Engineering, June 3-8, 2018, Prague, Czech Republic (accepted).
12. Porumb M., Castaldo, R., and Pecchia, L. (2018). Estimation of the heart rate variability features via Recurrent Neural Networks. In World Congress on Medical Physics and Biomedical Engineering, June 3-8, 2018, Prague, Czech Republic (accepted).

Book chapter

1. Pecchia, L., Castaldo, R., Melillo, P., Bracale, U., Craven, M., and Bracale, M. (2015). Early Stage Healthcare Technology Assessment. Clinical Engineering: From Devices to Systems, 95.

Abstract

Shifting healthcare monitoring techniques from laboratory into real-life scenarios has always been very challenging. The current shift towards the use of advanced sensors into everyday objects (e.g., smartwatches) is actively increasing the need for reliable methods and tools to analyse healthcare information acquired in real-life settings for wellbeing applications. In fact, the diffusion of wearable sensors has opened new and unexplored scenarios for Cardiovascular System (CVS) and Autonomic Nervous System (ANS) monitoring in real-life settings. As such, this thesis aims to develop methods and tools to monitor the relationship between CVS and ANS in real-life settings via biomedical signal processing and data-driven machine learning techniques, with the goal of predicting adverse healthcare events and automatically detecting the onset of unhealthy risky situations. Therefore, to investigate the relation between CVS and ANS, electrocardiogram signals and in particular Heart Rate Variability (HRV) were widely investigated in two case studies: acute mental stress detection and prediction of accidental falls in later-life via HRV.

One of the main limitations of using wearable sensors for the detection of risky situations in real-life settings is the need to shorten the length of physiological signals below the standard recommendations, which may cause a loss of accuracy in the detection of adverse healthcare events. Therefore, this problem was investigated taking as an exemplar mental stress detection, which is a cogent problem for modern society and it is well-known that mental stress causes alterations in both CVS and ANS. Through a systematic review of the literature, it was demonstrated that little attention has been paid thus far to ultra-short term HRV analysis (i.e., less than 5 minutes) for mental stress detection. Consequently, four experiments were designed and carried out in real-life and in-lab environments to propose a systematic method combining both statistical and machine learning methods to select ultra-short HRV features that are reliable surrogates of 5min HRV features. As a consequence, this study proved that it is possible to automatically detect real mental stress with 1min recordings achieving accuracy rate of 88%.

Another limitation of using wearable sensors is the need to improve machine learning techniques to enhance the prediction of rare events. In order to address this, an unbalanced dataset was investigated. In particular, a study was designed to apply data-driven machine learning techniques to an unbalanced dataset of ECG recordings acquired from 170 hypertensive elderly patients, of which 34 experienced an accidental fall. An experimental framework for data-driven machine learning techniques to detect rare events (i.e., falls) was developed to reduce the risk of overfitting problems in unbalanced datasets. This study was the first proving that short term HRV recordings could be used to identify future fallers with high accuracy.

This research achieved novel results and significant knowledge advancement for both the investigated well-being and health problems as well as methodological techniques.

List of Abbreviations

AE	Academic Examination.
AI	Artificial Intelligence.
ANNs	Artificial Neural Networks.
ANS	Autonomic Nervous System.
AR	Auto-Regressive.
AT	Arithmetic Task.
BMI	Body Mass Index.
BP	Blood Pressure.
CD	Correlation Dimension.
CI	Confidence Interval.
CVS	Cardiovascular System.
CWT	Computer Work Task.
DFA	Detrended Fluctuation Analysis.
ECG	Electrocardiogram.
FFT	Fourier Fast Transform.
FS	Flight Simulator.
HR	Heart Rate.
HRV	Heart Rate Variability.
IBK	K-Nearest Neighbour Classifier.
ICC	Intra-class Correlation Analysis.
ICT	Information and Communication Technology.

IoT	Internet of Things.
IQR	Interquartile Range.
LDA	Linear Discriminat Analysis.
LOO	Leave-one-out.
LT	Logic Task.
MLP	Multilayer Perceptron.
MNB	Multinomial Naïve Bayes.
MT	Memory Task.
NB	Naïve Bayes.
PMT	Physical-Mental Task.
PNS	Parasympathetic Nervous System.
PP	Poincaré Plot.
PSD	Power Spectrum Density.
PST	Public Speech Task.
RBF	Radial Basis Function.
ROC	Receiver Operating Characteristic.
RP	Recurrence Plot.
SA	Sinoatrial.
SCWT	Stroop Colour Word Test.
SNS	Sympathetic Nervous System.
SVM	Support Vector Machine.
VGC	Video Game Challenge.

Chapter 1

Introduction to the Research

1.1 Chapter overview

This chapter introduces the main aspects of the research. It guides the reader through the explanation of the research topic (section 1.2) and the rationale behind the research (section 1.3). The main research questions, aims and objectives are rigorously explained in this chapter (section 1.4). An overview of the methodologies used during the research is also provided (section 1.5). Lastly, a detailed structure of the entire thesis is presented to guide the reader through the research (section 1.6).

1.2 Introduction to the research topic

The rapid integration of Information and Communication Technology (ICT), Internet of Things (IoT) and Artificial Intelligence (AI) has introduced completely new scenarios in our lives. Until recently, continuous monitoring of physiological parameters was only possible for a short time and in controlled environments such as hospital and laboratory settings. Today, with developments in the field of wearable technologies, the possibility of continuous, real-time monitoring of physiological signals is a reality also in real-life settings (i.e., not hospitals or laboratories, but in daily-life activities) [12]. These advancements have opened up new opportunities in the prevention, timely diagnosis, control, treatment of diseases and monitoring in real-life settings. Nevertheless, the translation of signal processing and data mining techniques from controlled environments (i.e., hospitals and research laboratories) into real-life settings using wearable sensors has brought numerous challenges and theoretical limits. In particular, shortening biomedical signals below the standard re-

commendations (ultra-short recordings) could lead to erroneous analysis and therefore, unreliable information. Moreover, in the real world, some healthcare events (e.g., strokes, accidental falls) are rarely detected resulting in small and unbalanced datasets whose analysis via machine learning techniques are still challenging.

Therefore, the current shift towards the use of advanced sensors into everyday objects (e.g., smartwatches) has strongly increased the need for reliable methods and tools to analyse healthcare information acquired in real-life settings for wellbeing applications. In this thesis, novel approaches and tools are developed to reliably detect and predict adverse healthcare events in real-life settings overcoming some of the main challenges and theoretical limitations in dealing with real-life data. In particular, the major problems and theoretical limitations explored in this research are related to biomedical signal processing and machine learning techniques applied to real-life settings.

1.3 The rationale of the research

With the proliferation of innovative monitoring technologies, the interest in the interaction between the Autonomic Nervous System (ANS) and the Cardiovascular System (CVS) has grown exponentially as it has brought to light the opportunity to continually monitor this interaction using non-invasive biomarkers, providing significant information on the status of a subject. In fact, measures of autonomic function -such as Heart Rate Variability (HRV)- can yield relevant prognostic information. In the light of that, in this thesis, the interaction between the ANS and the CVS is widely explored via HRV analysis in challenging problems such as stress detection and fall prediction in later-life.

In fact, the emergence of factors such as the increase in elderly population and various chronic diseases (e.g., stress) has brought more emphasis on monitoring the interaction between the ANS and the CVS as a driver to predict adverse healthcare events, detect well-being problems and recognise primary risks for health.

As a matter of fact, recently, the research area of health monitoring systems has shifted from simple reasoning of wearable sensor readings (like calculating the sleep hours or the number of steps per day) to the higher level of data analysis in order to give the end-user much more information on illness or other factors that affect health (e.g., stress level).

However, the use of wearable devices collecting physiological signals in real-life settings has raised many challenges regarding the data analysis process. For instance, acquiring biomedical signals in time windows below the standard recom-

recommendations has brought new concerns about the validity and reliability of the signal analysis. Similarly, machine learning techniques present several limitations when applied to real-life data, for instance to predict rare events.

In light of this scenario, this research through experimental methods provides novel approaches and tools to overcome some of the main theoretical limitations mainly regarding signal processing and data-driven machine learning techniques applied to real-life data.

Different experiments were designed to develop methods and tools facilitating the prediction of adverse healthcare events and automatic detection of the onset of unhealthy, risky situations, monitoring the relationship between the CVS and the ANS via HRV analysis in real-life settings.

The methods and tools presented in this thesis will contribute to the improvement of healthcare wearable sensors.

1.4 Research questions, aim and objectives

The proliferation of wearable sensors (e.g., embedded in smartwatches or mobile phones) has opened further and unexplored scenarios for CVS and ANS monitoring in real-life settings. However, the use of wearable devices collecting physiological signals in real-life settings has raised new questions. In particular, this research explored and investigated the following questions:

Research Question 1: to what extent can the length of biomedical signals be shortened without losing their physiological meaning?

Research Question 2: how can current machine learning techniques be improved to reliably assess the interaction of the CVS and the ANS in real-life settings?

Therefore, the **main aim** of this research was to develop reliable and accurate methods and tools to monitor the relationship between the CVS and the ANS in real-life settings via biomedical signal processing and machine learning techniques to predict adverse healthcare events and automatically detect the onset of unhealthy risky situations.

Accordingly, the main objectives were to:

Objective 1 (Obj 1): develop a novel approach to assess the reliability of biomedical signals length shorter than the standard recommendations in real-life settings.

Objective 2 (Obj 2): develop a pragmatic framework to improve machine learning techniques for small datasets.

Objective 3 (Obj 3): develop a pragmatic framework to improve machine learning techniques for unbalanced datasets (i.e., reducing the number of false positive classifications and overfitting problems to predict rare events).

In order to fulfill these objectives and hence develop methods and tools to monitor the CVS and ANS relationship in real-life settings, the following case studies were explored and analysed:

Case Study 1 (CS1): monitoring of cardiovascular and autonomic response to mental stress in healthy subjects.

Case Study 2 (CS2): monitoring of cardiovascular and autonomic response to falls in later-life.

As a consequence, the deliverables set to investigate the two case studies and accomplish the aforementioned aim and objectives were:

1. for Case Study 1: “the cardiovascular and autonomic response to mental stress”
 - (a) to extract the most informative HRV features and study designs to highlight existing results and pave the way for empirical studies;
 - (b) to identify existing methods and tools to assess the reliability of biomedical signals shorter than standard recommendations (i.e., ultra-short HRV analysis, below 5 minutes);
 - (c) to investigate mental stress using shorter biomedical signals (i.e., ultra-short HRV analysis) in real-life settings;
 - (d) to carry out experiments demonstrating the relationship between shorter biomedical signal (i.e., ultra-short HRV analysis) and acute mental stress in laboratory settings;
 - (e) to explore and assess the power of real-life and in-lab stress;
 - (f) to develop an accurate machine learning algorithm for the detection of mental stress in healthy subjects;
2. for Case Study 2: “the cardiovascular and autonomic response to falls in later-life”

- (a) to identify the main risks of falls and the existing technologies and tools to predict falls in later-life;
- (b) to analyse the most informative HRV features to predict falls in later-life;
- (c) to develop a predictive algorithm using advanced machine learning techniques for predicting falls in later-life.

Table 1.1 summarises the explored case studies, objectives and specific deliverables.

Table 1.1: Summary of the explored case studies, objectives and deliverables.

Case studies	Objectives	Deliverables	My published work
CS1: mental stress detection	Obj1: method to assess ultra-short HRV	1.a: evidence-based study designs	[1], [2]*
		1.b: method selection to assess ultra-short HRV	[3]#
		1.c: analysis of ultra-short HRV in real-life	[5]#, [6]*, [7]*
		1.d: analysis of ultra-short HRV in-lab	[8]*, [9]*
		1.e: assessment of real VS in-lab stressors	
	Obj2: ML for small dataset	1.f: development of ML algorithm to detect mental stress	
CS2: prediction of falls in later-life	Obj3: ML for unbalanced dataset	2.a: identification of the main risks of falls and prevention programmes	[4], [10]*, [11]*
		2.b: analysis of short HRV features to predict falls	
		2.c: development of ML algorithm to predict falls	

ML: Machine Learning; * conference paper; #journal paper under review.

In order to fulfil the aforementioned objectives, five different experiments were designed and carried out:

Experiment 1 (E1): a dataset of 42 healthy subjects, with more than 7 hours of Electrocardiogram (ECG) recordings was analysed to investigate the reliability

of biomedical signals (i.e., HRV) shorter than the standard recommendations during real-life stress.

Experiment 2 (E2): 128 healthy subjects were enrolled and more than 22 hours of ECG recordings were analysed to demonstrate the relationship between shorter biomedical signals (i.e., ultra-short HRV analysis) and acute mental stress in-lab settings using a cognitive stressor (i.e., Stroop Colour Word Test, SCWT [13]).

Experiment 3 (E3): 42 healthy subjects were enrolled and more than 7 hours of ECG recordings were analysed to support the relationship between shorter biomedical signals (i.e., ultra-short HRV analysis) and acute mental stress in-lab settings using a different cognitive stressor (i.e., Video Game Challenge, VGC [14]).

Experiment 4 (E4): the cumulative number of subjects investigated in E1, E2 and E3 was analysed to explore the power of real and in-lab stressors.

Experiment 5 (E5): a dataset of 170 hypertensive subjects, with more than 340 hours of ECG recordings was analysed to identify the most informative HRV features for the prediction of falls in later-life and develop a predictive algorithm using advanced machine learning for unbalanced datasets.

In order to carry out these experiments, signal analysis of already acquired data and new acquisitions were executed as specified in Table 1.2.

Table 1.2: Signal analysis and acquisitions.

Signal Acquisitions & Analysis	N° of Sub.	Hours of ECGs	Populations
Analysis of real-life stress data	42	7 h	Healthy subjects
Acquisition and analysis of in-lab stress (SCWT) data	128	22 h	Healthy subjects
Acquisition and analysis of in-lab stress (VGC) data	42	7 h	Healthy subjects
Analysis of real-life fallers' data	170	340 h	Hypertensive patients

SCWT: Stroop Colour Word Test; VGC: Video Game Challenge; N°: Number; Sub.: Subjects.

1.5 Research methodology

The interest in monitoring the relationship between the CVS and the ANS has gained significant momentum in the recent years as the vast diffusion of new healthcare

technologies has made it possible to monitor vital signs continuously and therefore, predict adverse healthcare events and automatically detect the onset of unhealthy, risky situations in real-life settings. However, in order to decrease the computational complexity of the models implemented in wearable sensors and to detect adverse events in real-time, the length of biomedical signals needs to be shortened below the standard recommendations.

Among the different available non-invasive techniques for assessing the autonomic status, ECG and in particular HRV have emerged as a simple, non-invasive method to evaluate the sympathovagal balance at the sinoatrial level [15]. In support of that, it has been used in a variety of clinical situations [15]. As a consequence, in order to investigate the relationship between the CVS and the ANS, ECG and HRV were widely explored during this research. The identification of the major problems and theoretical limitations around biomedical signal processing and data-driven machine learning led me to define the main research questions and objectives of this research (section 1.4).

Theoretical limitations regarding the analysis of ultra-short signals (i.e., time horizon), and machine learning techniques for small and unbalanced datasets were investigated in two different case studies (i.e., mental stress detection and accidental fall prediction) as shown in Fig. 1.1. Several experiments were designed and carried out to develop robust methods and tools using advanced biomedical signal processing and machine learning techniques. The relationship between the CVS and the ANS was monitored to detect or predict adverse healthcare events in real-life settings.

Mental stress detection was appointed as an exemplar case study (CS1) to investigate the shortening of physiological signal length below standard recommendations (i.e., the time horizon) (Obj 1). In fact, mental stress detection was chosen as a case study not only because it is an important problem for the modern society and causes alterations in both the CVS and the ANS, but also because an increasing number of off-the-shelf wearable devices and apps are already using shorter signals for the detection of mental stress, which has been a growing topic over the years [16–20]. Moreover, through a systematic review of the literature (deliverables 1a and 1b), it was demonstrated that little attention has been paid thus far to ultra-short term HRV analysis (i.e., below 5 min) and that there were not existing methods or tools to reliably assess ultra-short HRV analysis for mental stress detection. Consequently, in order to fill those gaps and seek answers to the research questions reported in section 1.4, four experiments were designed and carried out: stress assessment in real life (E1) (deliverable 1c); stress assessment in individual cognitive tasks (i.e., Stroop Colour Test, E2); stress assessment in a group war scenario simulator (i.e.,

war rescue mission in competitive and challenging virtual gaming, E3) (deliverables 1d); the combination of both real and in-lab stressors (E4) to understand the power of real and in-lab data (deliverable 1e). Furthermore, stress assessment in real-life (E1) led to the refinement of a pragmatic framework for small datasets as the available data in real-life settings are often limited by the scarcity of good quality data (Obj 2). Consequently, a classifier to detect mental stress via ultra-short HRV features was also developed (deliverable 1f).

Regarding machine learning techniques and unbalanced datasets (Obj 3), fall prediction in later-life was appointed as exemplar case study (CS2), because accidental falls are one of the best examples of rare events. Through a review of the existing literature, it was clear that there was a huge gap not only in terms of theoretical frameworks investigating unbalanced datasets using wearable sensors during real life, but also in monitoring the CVS and the ANS relationship as a tool to predict falls in later-life (deliverable 2a). Therefore, an experiment (E5) was designed and carried out to apply data-driven machine learning techniques to a dataset of more than 340 hours of continuous ECG recordings, acquired from 170 hypertensive patients (mean age above 55), of which 34 experienced an accidental fall (defined as an unintentionally coming to the ground or some lower level, not due to syncope) within three months from recording after the baseline assessment. HRV features were analysed to identify the most informative features for the prediction of falls (deliverable 2b). A robust model to predict falls was also developed via HRV features (deliverable 2c).

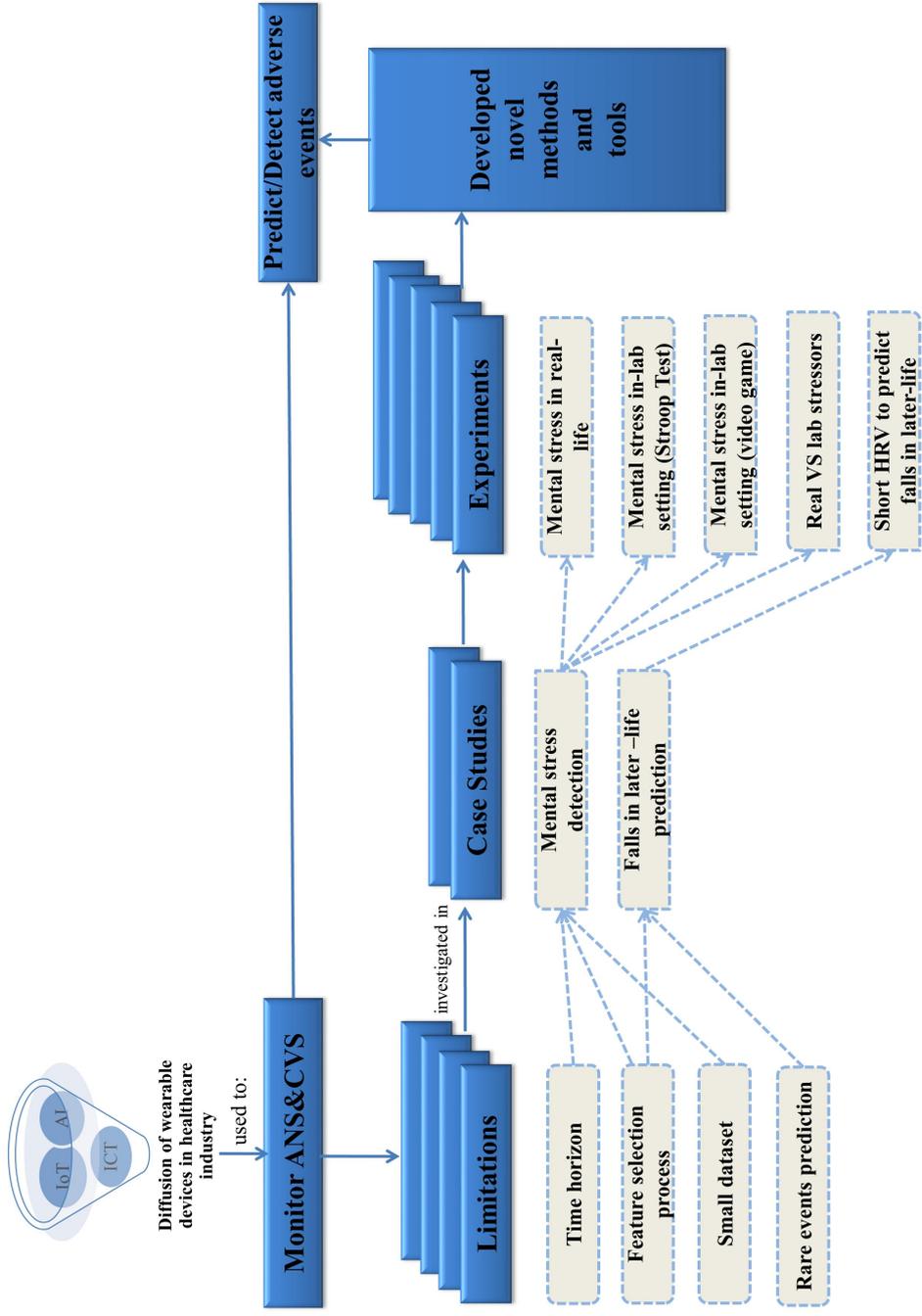


Figure 1.1: Research process. With the proliferation of wearable sensors, theoretical limitations have been raised regarding CVS and ANS monitoring in real-life settings. Some of these limitations (i.e., time horizon, machine learning techniques for small and unbalanced datasets) were investigated in two case studies (i.e., mental stress detection and accidental fall prediction). Several experiments were designed and carried out to develop robust methods and tools to detect or predict adverse healthcare events in real-life settings.

1.6 The structure of the thesis

Chapter 1: Introduction to the Research

This chapter provides background information about the growing interest around wearable technologies in the healthcare industry, opening new scenarios to monitor the interaction between the CVS and the ANS, but also new challenges and theoretical limits that need to be overcome. The focus of the research, the main aim and research objectives are also presented.

Chapter 2: Background

In this chapter the basic science about the interaction of the CVS and the ANS, the signal processing and data-driven machine learning techniques used in this thesis are explored.

Chapter 3: Literature Review on Acute Mental Stress and Falls in Later-life

Chapter 3 presents the state-of-the-art for mental stress detection and fall prediction in later-life via HRV. In particular, the first part of this chapter systematically reviewed the main HRV features and state-of-the-art study designs to detect mental stress. Moreover, a review of the existing methods to assess ultra-short HRV is also presented. The second part introduces a brief overview of the existing main fall risks, prevention programmes, the existing technologies and tools to predict falls in later-life. Furthermore, the relationship between the risk of falling and HRV is also discussed.

Chapter 4: Development of Methods and Tools to Monitor Cardiovascular and Autonomic Response in Real-life Settings

Chapter 4 presents the methods and tools developed to monitor the CVS and ANS in real-life settings. The first part of this chapter is focused on a novel framework to investigate to what extent biomedical signals (i.e., HRV) can be shortened (i.e., < 5min) without losing important physiological information. In other words, whether HRV features extracted from ultra-short excerpts (i.e., < 5min) can be considered surrogates of short term HRV features. In fact, this is the most prominent requirement for wearable sensors to detect or predict adverse healthcare events in quasi-real-time. The second part of this chapter is focused on defining pragmatic frameworks to improve machine learning techniques in order for them to cope with small and unbalanced datasets. In fact, another important issue for establishing a reliable supervised learning strategy in real-life settings and preventing over-fitting problems is to properly make use of the available samples, especially when the number of available samples is small or when one or more classes occur far less frequently than others.

The main objectives defined in Chapter 1 are tackled here.

Chapter 5: Cardiovascular and Autonomic Response to Mental Stress

Chapter 5 discusses the cardiovascular and autonomic response to stress, exploring mental stress in real-life and also in-lab environments. The method to assess the reliability of shorter signals is applied here. In fact, this study proves that not all the ultra-short term HRV features are good surrogates of short term ones. In fact, only six ultra-short term HRV features resulted to be good surrogates of short term ones. Moreover, an automatic classifier to detect acute mental stress is also presented. The automatic classifier is able to detect stressed subjects with very high performances, using 3 min HRV analysis, and relatively good performances using 1 min HRV excerpts.

Chapter 6: Cardiovascular and Autonomic Response to Falls in Later-life

Chapter 6 explores the cardiovascular and autonomic response to falls in later-life. As opposed to the few previous studies investigating HRV in fallers, which were focused on 24-hour HRV analysis, this is the first study describing the results obtained with short term HRV analysis, which is much easier and cheaper to be translated into everyday outpatient clinical practice. This approach is based on the idea that it is possible to detect constantly depressed ANS status early, which increases the risk of falling significantly. Moreover, the theoretical framework developed to predict rare events is applied in real-life settings using wearable devices, enabling the prediction of fallers with a sensitivity rate of 72% and a specificity rate of 61%.

Chapter 7: Conclusions and Future Work

Chapter 7 reports the major conclusions of the research; it addresses how far the aims and objectives of this thesis have been met and limitations arising from the research. Additionally, recommendations are suggested for future work.

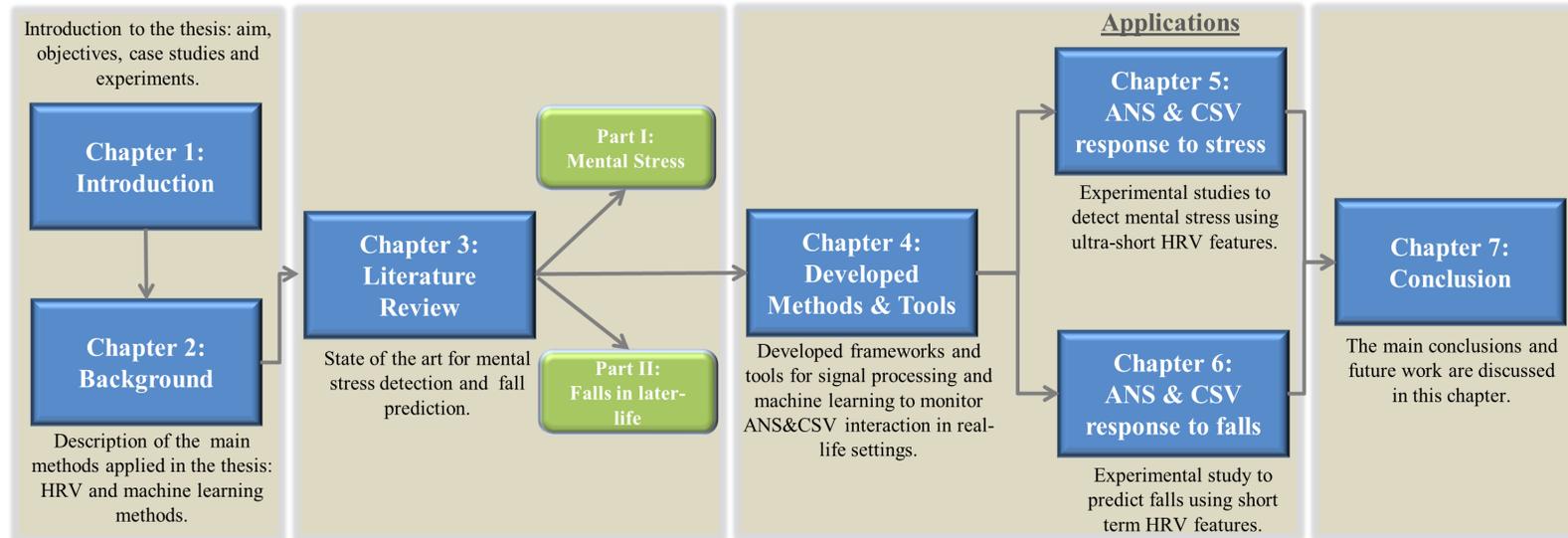


Figure 1.2: The thesis at a glance.

Chapter 2

Background

2.1 Chapter overview

This chapter provides background information of relevance to this thesis. It comprises a brief overview of the basic science about the interaction of the CVS and the ANS (section 2.2), a detailed description of HRV analysis and the factors influencing it (subsection 2.2.1). This chapter also presents a short review of the theoretical limitations and challenges around biomedical signals processing and the development of machine learning algorithms for the detection and prediction of adverse healthcare events in real-life settings (section 2.3). Finally, the signal processing and data-driven machine learning techniques applied in this thesis are described in section 2.4.

2.2 Cardiovascular and Autonomic Nervous Systems

In 1628, for the first time, William Harvey wrote about a link between the brain and the heart saying: “*for every affection of the mind that is attended with either pain or pleasure, hope or fear, is the cause of an agitation whose influence extends to the heart*” [21]. For the past century, numerous studies have investigated the link between the CVS and the ANS, finding it to be very complex [22].

The CVS has two primary components: the heart and blood vessels [23]. The cardiovascular system is subject to precise reflex regulation to supply and reliably provide oxygenated blood to different body tissues under a wide range of circumstances. The activity of cardiovascular control is largely regulated by the ANS [24].

The ANS is part of the peripheral nervous system and controls the function of many muscles, glands and organs within the body. The autonomic system functions

in a reflexive and involuntary manner [25]. The role of the ANS is to constantly fine-tune the functioning of organs and organ systems according to both internal and external stimuli. It helps to maintain homeostasis through the coordination of various activities such as hormone secretion, circulation, respiration, digestion, and excretion [25]. The ANS is subdivided into two separate divisions: the Sympathetic Nervous System (SNS) and the Parasympathetic Nervous System (PNS). Keeping in mind that both systems work in synergy to maintain homeostasis within the body, it is important to understand how these two systems function in order to determine how they each affect the body and the CVS. Both the sympathetic and parasympathetic nerves release neurotransmitters, primarily norepinephrine and epinephrine for the SNS, and acetylcholine for the PNS. These neurotransmitters (also called catecholamines) relay the nerve signals across the gaps (synapses) created when the nerve connects to other nerves, cells or organs. The neurotransmitters then attach to either sympathetic receptor sites or parasympathetic receptor sites on the target organ to exert their effect. This is a simplified version of how the ANS functions (as shown in Fig. 2.1).

The SNS is commonly known as the “fight or flight” response, which sees the activation of adrenergic receptors:

- increased sweating;
- decreased peristalsis;
- increased heart rate (increased conduction speed, decreased refractory period);
- pupil dilation;
- increased blood pressure (increased contractility, increased the ability of the heart to relax and fill).

The PNS is sometimes referred to as the “rest and digest” system. In general, the PNS acts in the opposite way to the SNS, reversing the effects of the fight or flight response. However, it may be more correct to say that the SNS and the PNS have a complementary relationship, rather than one of opposition.

The activation of the PNS is seen in:

- decreased sweating;
- increased peristalsis;
- decreased heart rate (decreased conduction speed, increased refractory period);

- pupil constriction;
- decreased blood pressure (decreased contractility, decreased the ability of the heart to relax and fill).

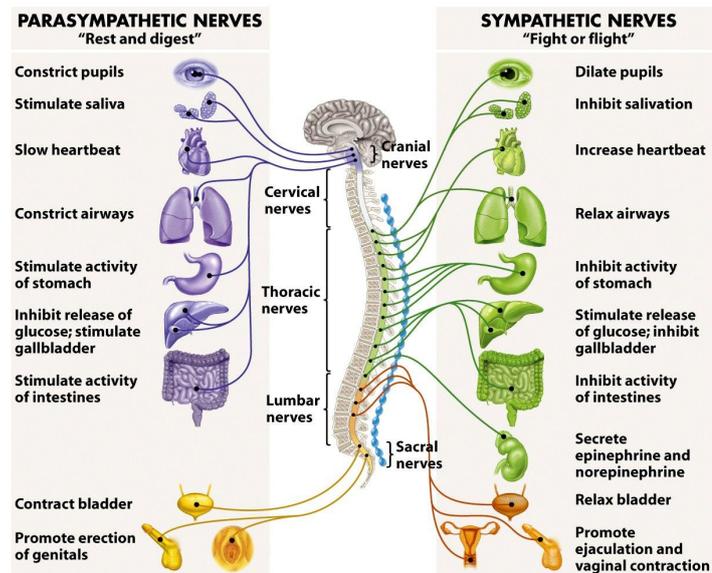


Figure 45-20 Biological Science, 2/e
© 2005 Pearson Prentice Hall, Inc.

Figure 2.1: Sympathetic and Parasympathetic branches [26]. The ANS is subdivided into two separate divisions: the SNS and the PNS. Both systems work in synergy to maintain homeostasis within the body. In general, the PNS acts in the opposite way to the SNS.

The ANS affects changes in the body that are meant to be temporary; in other words, the body should return to its baseline state. It is natural that there should be brief excursions from the homeostatic baseline, but the return to baseline should occur in a timely manner. When one system is persistently activated (increased tone), health may be adversely affected. In fact, there are numerous diseases and conditions which result from ANS and CVS dysfunction, (e.g., orthostatic hypotension, acute and chronic stress, cardiovascular diseases, cognitive dysfunction, Parkinson's disease) [27]. Therefore, being able to monitor the cardiovascular and autonomic interaction opens new scenarios for the prediction of adverse healthcare events or for the detection of the onset of unhealthy situations in real-life settings.

2.2.1 Heart Rate Variability

Although there are several techniques applicable to the field of autonomic and cardiovascular monitoring, one of the most reliable and non-invasive tools, already in-use in wearable sensors, is Heart Rate Variability (HRV) [15]. HRV is mainly extracted from an ECG, which is one of best known and deeply analysed biomedical signals [28, 29]. Therefore, in this thesis, HRV was selected as the primary biomedical signal to investigate the translation of biomedical signal processing and machine learning techniques from the lab in real-life settings.

HRV describes the variations between consecutive heartbeats (Fig. 2.2) [15, 30].

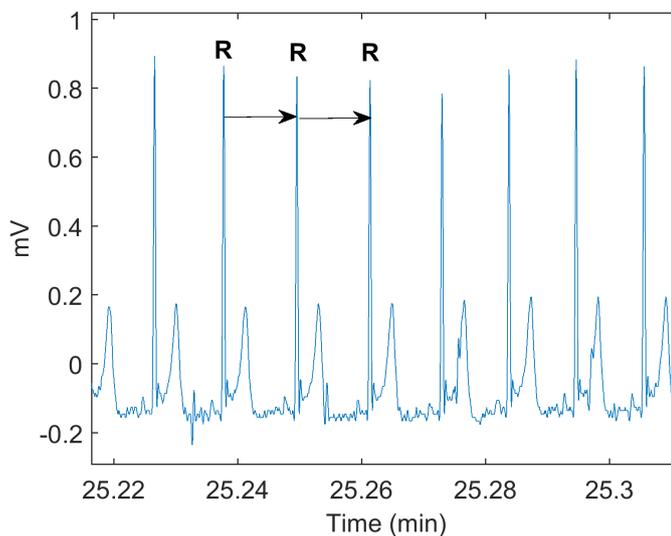


Figure 2.2: Raw ECG trace (from one of the studies) depicting the difference in duration between heart beats (R wave to R wave intervals).

The rhythm of the heart is controlled by the Sinoatrial (SA) node, which is modulated by both the SNS and the PNS [31]. Sympathetic activity tends to increase heart rate and its response is slow (few seconds). Parasympathetic activity, on the other hand, tends to decrease heart rate and mediates faster (0.2 to 0.6 seconds). The continuous modulation of the sympathetic and parasympathetic innervations results in variations in venous volume ΔV_v , left ventricular contractility V_c and the Heart Rate (HR) (Fig. 2.3) [32]. Both the SNS and the PNS are constantly monitored by baroreceptors, which are located on the walls of some large vessels and can sense the increase in Blood Pressure (BP) caused by the stretching of the vessel walls. In fact, the baroreceptors convert the sensory environment into neural sig-

nals, which are integrated by the SNS and the PNS generating afferent information to activate a response (Fig. 2.3). In particular, the efferent impulses innervate the heart and blood vessels, causing an increase in HR and in total peripheral resistance (TPR), which contributes to an increase in the arterial blood pressure (BP_a) [33].

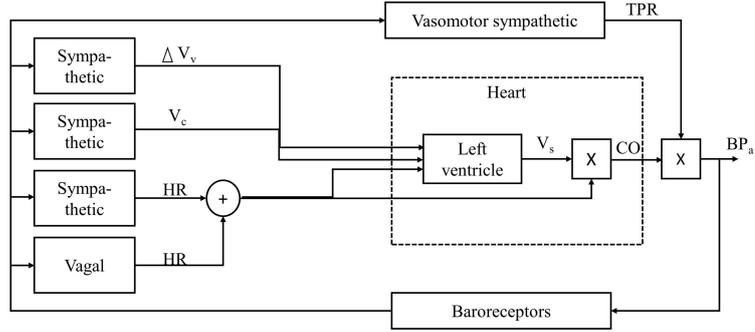


Figure 2.3: Diagram of the Cardiovascular Control System. The modulation of the sympathetic and parasympathetic (vagal) innervations result in variations in venous volume (ΔV_v), left ventricular contractility (V_c) and HR. The efferent impulses, generated by baroreceptors, innervate the heart and blood vessels, causing an increase in HR and in total peripheral resistance (TPR), which contribute to an increase in the arterial blood pressure (BP_a). V_s : stroke volume; CO: cardiac output.

HRV analysis examines the sinus rhythm modulated by the ANS [30, 31]. Therefore, one should technically detect the occurrence times of the SA-node action potentials. This is, however, practically impossible and, thus, the fiducial points for the heart beat are usually determined from an ECG recording. The nearest observable activity in the ECG compared to SA-node firing is the P-wave resulting from atrial depolarization (Fig. 2.4) and, thus, the heart beat period is generally defined as the time difference between two successive P-waves. The signal-to-noise ratio of the P-wave is, however, clearly lower than that of the strong QRS complex which results primarily from ventricular depolarization. Therefore, the heart beat period is commonly evaluated as the time difference between the easily detectable QRS complexes termed the R-R or N-N (normal-normal) intervals. The N-N intervals are all of the intervals between adjacent QRS complexes resulting from sinus node depolarizations [15]. In this thesis ECG recordings were free of ectopic and missing data, therefore, the R-R intervals corresponded to the N-N intervals. HRV is the measurement of the variability of the N-N intervals [15].

The accuracy of HRV analysis may be affected by the sampling rate at which the ECG is digitalised and measurement noise (e.g., electrode motion, contact noise,

electromiography noise) [28]. Therefore, in order to minimise these errors caused by the use of unstandardised ECG equipment or incorrectly used techniques to extract HRV features, the minimum sampling frequency rate at which the ECG is digitalised should be of least 250 Hz and any measurement noise should be properly filtered following the standard guidelines presented by the ESC/NASPE Task force guidelines for HRV [15]. More details are given in section 2.4.1.

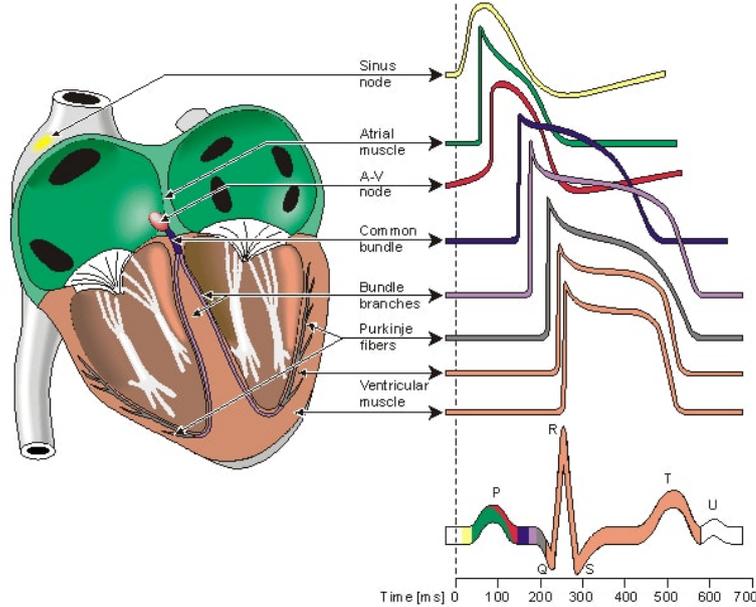


Figure 2.4: Electrophysiology of the heart [33]. Initial activation of the sinus atrial node (SA) is followed by atrial depolarisation and contraction, and conduction to the atrio-ventricular (AV) node. From the AV-node, the action potential is conducted through the common bundle and Purkinje fibres to the ventricles, followed by ventricular depolarisation and contraction. Finally, the ventricles relax in the repolarisation phase.

2.2.1.1 HRV Analysis

HRV parameters can be analysed through different domains such as time, frequency, time-frequency domains and a non-linear domain. HRV analysis can be performed on 24 hour nominal recordings (defined as long term HRV analysis), 5 minute recordings (defined as short term HRV analysis) or even shorter recordings. In this thesis, ultra-short term HRV analysis is defined as the analysis performed on HRV excerpts shorter than 5 minutes.

Linear time domain feature analysis Linear time-domain features were standardised in 1996 by the ESC/NASPE Task force guidelines for HRV features [15]. The time domain analysis is a straightforward evaluation, but the time features do not show high discrimination between sympathetic and parasympathetic contributions.

The time domain methods are the simplest to perform since they are applied directly to the series of successive NN interval values. The most evident measure is the mean value of NN intervals (MeanNN) or, correspondingly, the mean HR (MeanHR). In addition, several variables that measure the variability within the NN series are also computed. The standard deviation of RR intervals (StdNN) is defined as:

$$\text{StdNN} = \sqrt{\frac{1}{N-1} \sum_{j=1}^N (NN_j - \bar{NN})^2} \quad (2.1)$$

where NN_j denotes the value of the j 'th NN interval and N is the total number of successive intervals. The StdNN reflects the overall (both short-term and long-term) variation within the NN interval series, whereas the standard deviation of successive NN interval differences (SDSD) given by:

$$\text{SDSD} = \sqrt{E\{\Delta NN_j^2\} - E\{\Delta NN_j\}^2} \quad (2.2)$$

can be used as a measure of the short-term variability. For stationary NN series $E\Delta NN_j = ENN_{j+1} - ENN_j = 0$ and SDSD equals the root mean square of successive differences (RMSSD) given by:

$$\text{RMSSD} = \sqrt{\frac{1}{N-1} \sum_{j=1}^{N-1} (NN_{j+1} - NN_j)^2} \quad (2.3)$$

Another measure calculated from successive NN interval differences is the NN50, which is the number of successive intervals differing by more than 50 ms or the corresponding relative amount:

$$\text{pNN50} = \frac{NN50}{N-1} * 100 \quad (2.4)$$

In addition to the above statistical measures, there are some geometric measures that are calculated from the NN interval histogram. The HRV triangular index is obtained as the integral of the histogram (i.e., the total number of NN intervals) divided by the height of the histogram which depends on the selected bin width. To obtain comparable results, a bin width of 1/128 seconds is usually recommended.

Another geometric measure is the TINN which is the baseline width of the NN histogram evaluated through triangular interpolation. However, the major disadvantage of this measure is the need for a reasonable number of NN intervals to construct the geometric pattern. In practice, recordings of at least 20 min (but preferably 24 hours) should be used to ensure the correct performance of the geometric methods, i.e., the current geometric methods are inappropriate for the assessment of short term changes in HRV.

Time domain features are reported in Table 2.1.

Table 2.1: HRV features in the time domain.

Features in time domain	Units	Description and interpretation
Statistical measures		
MeanNN	ms	The mean of NN intervals
StdNN	ms	Standard deviation of all NN intervals
MeanHR	1/min	The mean heart rate
StdHR	1/min	Standard deviation of instantaneous heart rate values
RMSSD	ms	The square root of the mean of the sum of the squares of differences between adjacent NN intervals
SDNN index	ms	Mean of the standard deviations of all NN intervals for all 5 min segments of the entire recording
SDSD	ms	Standard deviation of differences between adjacent NN intervals
NN50 count	ms	Number of pairs of adjacent NN intervals differing by more than 50 ms in the entire recording. Three variants are possible counting all such NN intervals pairs or only pairs in which the first or the second interval is longer
pNN50	%	NN50 count divided by the total number of all NN intervals
Geometrical features		
HRV triangular index	-	Number of normal NN intervals divided by the height of the histogram of all the normal NN intervals measured on discrete scale with bins of 1/128s (7.8125ms)
TINN	ms	Baseline width of the minimum square difference of triangular interpolation of the highest peak of the histogram of all normal NN intervals

Linear frequency domain features Frequency domain features are also described and approved in the HRV guidelines of 1996 [15]. In the frequency domain methods, a Power Spectrum Density (PSD) estimate is calculated for the NN interval series. The regular PSD estimators implicitly assume equidistant sampling and, thus, the NN interval series is converted to an equidistantly sampled series by

interpolation methods prior to PSD estimation. However, almost all of the published PSD estimation techniques described in the relevant literature require evenly sampled data. Pre-processing of the NN tachogram with re-sampling techniques (such as linear or cubic spline re-sampling) is usually the means of producing an evenly sampled time series. Re-sampling introduces an implicit assumption about the form of the underlying variation in the NN tachogram; cubic spline techniques are often used and assume that the variation between beats can be modelled accurately by a cubic polynomial. The frequency features are extrapolated from an HRV spectrum, which can be estimated using several methods. Methods for calculating PSD estimates may be divided into non-parametric and parametric methods. Four methods to estimate the HRV spectrum are the most diffused in the literature, two parametric methods: a Welch periodogram [34] and AR methods [35], and two non-parametric methods: Fourier Fast Transform (FFT) [36] and Lomb-Scargle periodogram [37]. The two most common methods for frequency-domain HRV metric estimation are AR spectral estimation and Fourier techniques. It is recommended using both parametric and non-parametric assessments when evaluating frequency domain HRV features [15]. However, in non-parametric FFT based method algorithms power spectral estimates are calculated by integrating the spectrum over frequency bands computing the discrete Fourier transform (DFT) [38] of a sequence, or its inverse, therefore, in order to obtain the required frequency resolution, the main requirement is long term record of data. The FFT suffers from spectral leakage effects due to windowing that are inherent in finite length data records [39]. Whereas, in a parametric AR model based power, spectrum estimation methods avoid the problem of leakage and provide better frequency resolution than the FFT [15, 35, 40].

Therefore, while the non-parametric methods have the advantage of algorithmic simplicity and rapidity, the parametric methods produce smoother spectral components that can be distinguished more easily, and if the model order is chosen correctly (usually of the order of 16) this can allow an accurate estimation of the PSD over very short windows [41]. This is the reason why the parametric (AR model based) approach is selected in this research as ultra-short records (≤ 5 min.) are explored. An example of a PSD using AR methods is shown in Fig. 2.5.

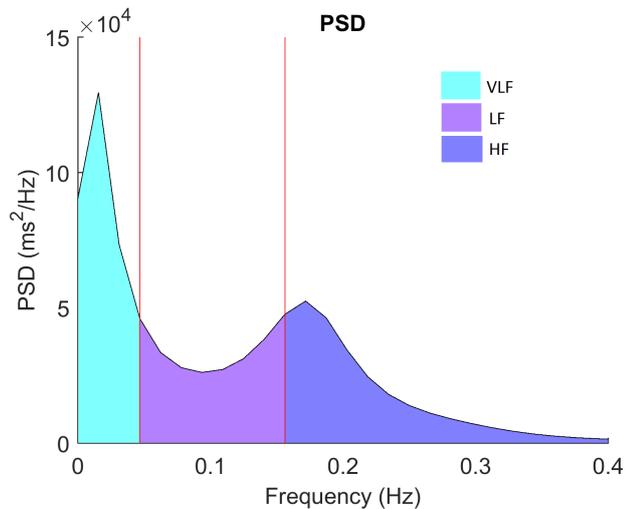


Figure 2.5: PSD estimation using AR method from a 5 min excerpt of a healthy subject under resting conditions. It represents the Very Low Frequency (VLF, 0.03-0.04 Hz), the Low Frequency (LF, 0.04-0.15 Hz) and High Frequency (HF, 0.15-0.4 Hz) band spectra.

The most common features used to characterise HRV spectrum are reported in Table 2.2.

Table 2.2: HRV features in the frequency domain.

Features in frequency domain	Units	Description and interpretation
ULF	ms ²	Ultra low frequency power (less than 0.03 Hz)
VLF	ms ²	Very low frequency power (between 0.03 and 0.04Hz)
LF	ms ²	Low frequency power (between 0.04 and 0.15 Hz)
HF	ms ²	High frequency power (between 0.15 and 0.40Hz)
LFnu	nu	Normalized low frequency power (LF/HF+LF)
HFnu	nu	Normalized high frequency power (HF/LF+HF)
LF/HF	-	Ratio of the low to high frequency power
TotPow	ms ²	The sum of the four spectral bands, LF, HF, ULF and VLF
Peak Frequency	Hz	LF, and HF band peak frequencies

The HF band is generally interpreted as an index of vagal modulation, while the LF band of both sympathetic and parasympathetic activity [42]. Despite some controversy, the ratio of the HF to LF absolute power (HF/LF) has been recurrently used as an index to describe the global instantaneous balance between sympathetic and vagal nerve activities (i.e., the so-called sympathovagal balance) [15]. The representation of LF and HF in normalised units (n.u.) emphasises the controlled and balanced behaviour of the two branches of the autonomic nervous system. Moreover, normalisation tends to minimise the effect on the values of LF and HF components

of the changes in total power [15]. Overall, frequency indices are used in both short term (i.e., 5 minutes) and long term ECG recordings (i.e., 24 hour Holter monitoring, with sampling frequency up to 1000 Hz). However, some frequency HRV features can also be analysed in shorter recordings. In fact, records should last for at least 10 times the wavelength of the lower frequency bound of the investigated component, and, to ensure the stability of the signal, should not be substantially extended. Thus, recordings of approximately 1 min are needed to assess the HF components of HRV while approximately 2 min are needed to address the LF component. Occasionally ultra-low frequencies (ULF; < 0.003 Hz) can also be used to analyse long-term acquisitions [15].

Time-frequency features HRV can also be analysed considering both time-domain and frequency domain features and the best results for HRV analysis are reported by discrete wavelet transform approach [43]. The time-frequency analysis method provides instantaneous and continuous assessment of HRV during stationary as well as transition phases of the N-N interval signal. However, power computation and implementation of those techniques may be problematic when applied to real-life setting [44]. In fact, these techniques could lose important information regarding the sympathetic and parasympathetic activations. Therefore, this approach is not used in this research.

Non-linear features Non-linear techniques are able to describe biological processes in a more effective way and they are more sensitive than linear during short term [45]. The use of non-linear measures represents a growing approach to HRV analysis. Differently, from linear indices, they are not influenced by non-stationarity of the signals. For this reason, they are well apt to appreciate how HRV reflects a chaotic system, like the heartbeat, which is dynamic, non-linear, and rapidly adjust over time.

Nonlinear properties of HRV are analysed by the following methods: Poincaré Plot (PP) [46], Approximate Entropy (ApEn) [47], Correlation Dimension (CD) [48], Detrended Fluctuation Analysis (DFA) [49], and Recurrence Plot (RP) [50].

Poincaré Plot The Poincaré Plot (PP) [46] is a common graphical representation of the correlation between successive NN intervals, for instance, the plot of NN_{n+1} versus NN_n . A widely used approach to analyse the Poincaré plot of an NN series consists of fitting an ellipse oriented according to the line-of-identity and computing the standard deviation of the points perpendicular to and along the

line-of-identity referred as SD1 and SD2, respectively (Fig. 2.6).

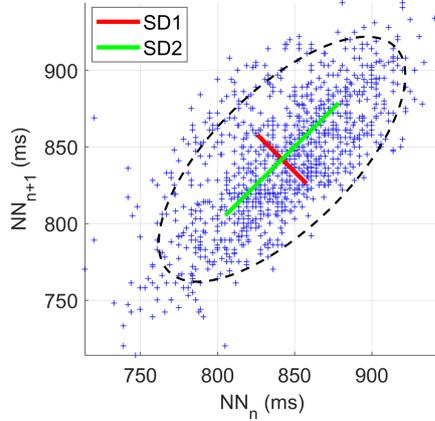


Figure 2.6: Example of Poincaré plot features for a healthy subject under resting condition. It is a common graphical representation of the correlation between successive NN intervals. SD1 and SD2 are the standard deviation of the points perpendicular to and along the line-of-identity.

These measures can offer useful physiological interpretations as regards ANS inferences: SD1 and SD2 values decrease following sympathetic stimulation with a concomitant change of shape in the plot. That is, the points are more scattered when vagal activity offsets the sympathetic one. At its simplest, a Poincaré plot offers a quick visual representation in order to understand the dynamics of a heartbeat data, thereby helping their understanding and supporting the derivation of inferences.

Entropy Entropy [47] measures the complexity or irregularity of the RR/NN series. Generally speaking, the term entropy describes the quantity of disorder in a system. When applied to a sequence, like the heart rate time series, it quantifies its regularity by averaging the information available. That is, the entropy rate indicates the level of entropy in a sequence when the sequence grows: if the entropy rate falls the process can be interpreted as regular and predictable, and vice versa. Entropy is a useful index for mapping heart rate fluctuations, because it requires relatively few data points. Moreover, several approaches have been proposed to evaluate both short- and long-length heartbeat interval series. One of the most popular is Approximate Entropy (ApEn), which measures the degree of irregularity within a series of data and is recommended when long and noise-free recordings are difficult to gather. ApEn shows the probability that similar configurations do not repeat. Assuming a generic time-series (i.e., x_1, x_2, \dots, x_N) and the inter-beat (NN) interval time series (NN_1, NN_2, \dots, NN_N). The algorithm to compute the ApEn of a NN time series

is developed as follows. A series of vectors of length m ($X_1, X_2, \dots, X_{N-m+1}$) is constructed from the NN intervals as $X_i = [NN_i, NN_{i+1}, \dots, NN_{i+m-1}]$. The distance $d[X_i, X_j]$ between the vectors X_i and X_j is defined as the maximum absolute difference between their respective scalar components (Eq. 2.5).

$$d[X_i, X_j] = \max_a |X_i(a) - X_j(a)|, \text{ where } X(a) \text{ is the } m \text{ scalar vector of } X. \quad (2.5)$$

For each vector X_i , the relative number of vectors X_j for which $d[X_i, X_j] \leq r$, $C_i^m(r)$ is computed; r is referred to as a tolerance value and is given by:

$$C_i^m(r) = \frac{\text{number of } d[X_i, X_j] \leq r}{N - m + 1}, \forall j \quad (2.6)$$

Then, the index $\Theta^m(r)$ is computed by taking the natural logarithms of each $C_i^m(r)$ and averaging them over i , namely:

$$\Theta^m(r) = \frac{1}{N - m + 1} \sum_{i=1}^{N-m+1} \ln C_i^m(r) \quad (2.7)$$

Finally, the approximate entropy is calculated as:

$$ApEn(m, r, N) = \Theta^m(r) - \Theta^{m+1}(r) \quad (2.8)$$

Several clinical studies [51, 52] have shown that either $m=2$ or an r between 0.1 and 0.2 times the StdNN are suitable to provide a reliable value of ApEn.

It is important to note that smaller values of ApEn indicate more frequent fluctuations and greater system regularity (i.e., the system is defined as deterministic), whereas, greater values convey more randomness and system complexity (i.e., the system is defined as random). The main advantage of this method is the opportunity to assess interactions between distinct systems, such as circadian rhythms and HRV. Yet, ApEn is sensitive to biases given by small trends in the data. Moreover, because it relies on a reference threshold to compute the similarity in the data, it may suggest greater regularity than what might actually be present in the recorded signals.

Another measure of entropy, Sample Entropy (SampEn), tackles this issue and provides a reliable estimate of the complexity of a signal, in particular for short series. While it holds the same overall meaning of ApEn, SampEn depends more on the given threshold, because it decreases as the threshold increases. The three

steps used to compute SampEn are described by the following formulae:

$$C_i^m(r) = \frac{\text{number of } d[X_i, X_j] \leq r}{N - m + 1} \forall j \neq i \quad (2.9)$$

$$\Theta^m(r) = \frac{1}{N - m + 1} \sum_{i=1}^{N-m+1} \ln C_i^m(r) \quad (2.10)$$

$$\text{SampEn}(m, r, N) = \log \frac{\Theta^m(r)}{\Theta^{m+1}(r)} \quad (2.11)$$

SampEn does not depend on N (the number of samples) but relies on the choice of the parameters m and r , as for ApEn. However, the dependence on the parameter r here is different: SampEn decreases when r increases. Note that with a high value of N and r , SampEn and ApEn often provide comparable results.

Other forms of entropy often used in HRV analysis are the Lempel-Ziv Entropy [53], which estimates the numbers of different and repeating patterns and generates a binary sequence, and multi-scale entropy (MSE) [54], which computes any entropy measure mentioned thus far for each time series, and displays them as a function of the number of data point in the period examined. However, they are not commonly used.

Physiologically, it is worth highlighting that the ANS modulates the heart rate to the constantly changing needs of an individual, and therefore, the heartbeat series is irregular with high entropy. However, when the cardiovascular system becomes less responsive to internal or external stimuli, entropy decreases and the time series becomes more ‘ordered’. That is, entropy indices progressively decrease during sympathetic activation and can, therefore, offer a more sophisticated appraisal of the sympathovagal balance and related inferences.

Correlation Dimension The correlation dimension [48] D2 is another method for measuring the complexity in HRV time series. As for Approximate Entropy, the series X_i is constructed and $C_i^m(r)$ is computed as in equation 2.9, but the distance function, in this case, is defined as the Euclidean distance (Eq. 2.12).

$$d[X_i, X_j] = \sqrt{\sum_{k=1}^m (X_i(k) - X_j(k))^2} \quad (2.12)$$

where, $X_i(k)$ and $X_j(k)$ refer to the k -th element of the series X_i and X_j , respectively. Hence, the following index $C^m(r)$ is computed by averaging $C_i^m(r)$ over i .

$$C^m(r) = \frac{1}{N - m + 1} \sum_{i=1}^{N-m+1} C_i^m(r) \quad (2.13)$$

The correlation dimension D2 is defined as the following limit value:

$$D2(m) = \lim_{r \rightarrow 0} \lim_{N \rightarrow \infty} \frac{\log C^m(r)}{\log(r)} \quad (2.14)$$

In practice this limit value is approximated by the slope of the regression curve ($\log(r)$, $\log(C^m(r))$). In the current research a value of $m = 10$ is adopted according to the existing literature [33, 45].

Detrended Fluctuation Analysis Detrended Fluctuation Analysis (DFA) measures the correlation within the signal [49]. DFA is a method developed to differentiate between the internal variations generated by complex systems, such as heartbeats and those variations caused by environmental or external stimuli. DFA enables characterisation of the internal correlations of the HRV signal as a function of a correlation distance. In other words, it allows quantification of the non-stationary heart rate signal by measuring how its variance is affected by the length of the heartbeat series. DFA is calculated for several segment lengths, and it increases when the segment length increases. For this reason, it can be used both for short- and long-term recordings.

DFA features are calculated following the steps described below.

1. The average \bar{NN} of the NN interval series is calculated on all the N samples. The alternate component of NN interval series, which is defined as NN minus its average value \bar{NN} , is integrated:

$$y(k) = \sum_{j=1}^k (NN_j - \bar{NN}) \quad k=1, \dots, N \quad (2.15)$$

2. The integrated series is divided into non-overlapping segments of equal length n . A least squares line is fitted within each segment, representing the local trends with a broken line. This broken line is referred as $y_n(k)$, where n denotes the length of each segment.
3. The integrated time series is detrended as follows: $y(k) - y_n(k)$. The root-mean-square fluctuation of the detrended time series is computed according

to the following formula:

$$F(n) = \sqrt{\frac{1}{N} \sum_{k=1}^N (y(k) - y_n(k))^2} \quad (2.16)$$

4. The steps from 2 to 4 are repeated for n from 4 to 64.
5. Representing the function $F(n)$ in a log-log diagram (Fig. 2.7), two parameters are defined: short-term fluctuations (dfa1) and long-term fluctuations (dfa2) as the slopes of the regression line relating $\log(F(n))$ to $\log(n)$. HRV scale-invariance has commonly been observed over a wide range, with a characteristic break at scales around 16 heart beats. Consequently, dfa1 and dfa2 are computed in the ranges of 4–16 and 16–64 n heart beats. Thus, these two fixed a priori defined scaling ranges are commonly assessed in HRV analysis [49, 55].

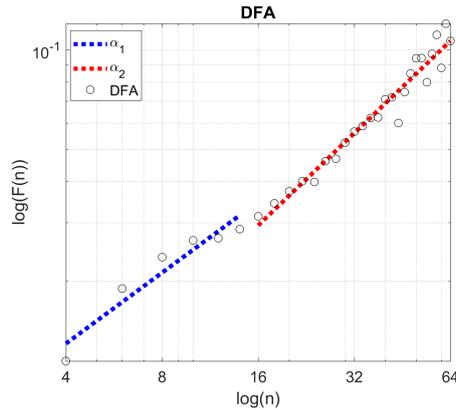


Figure 2.7: DFA example for a healthy subject during resting conditions. It measures the correlation within the signal through short-term fluctuations (in blue) and the long-term fluctuations (in red).

Recurrence Plot Recurrence Plot (RP) is another approach performed for measurement of the complexity of a time-series [50]. RP is designed according to the following steps.

Vectors $X_i = (NN_i, NN_{i+\tau}, \dots, NN_{i+(m-1)\tau})$, with $i = 1, \dots, k$, with $k = [N - (m-1) * \tau]$, where m is the embedding dimension and τ is the embedding lag, are defined.

A symmetrical k -dimensional square matrix M_1 is calculated computing the Euclidean distance of each vector X_i from all the others.

After choosing a threshold value r , a symmetric k -dimensional square matrix M_2 is calculated as the matrix whose elements $M_2(i, j)$ are defined as:

$$M_2(i, j) = \begin{cases} 1 & \text{if } M_1(i, j) < r \\ 0 & \text{if } M_1(i, j) > r \end{cases} \quad (2.17)$$

The RP is a representation of the matrix M_2 as a black (for ones) and white (for zeros) image as shown in Fig. 2.8.

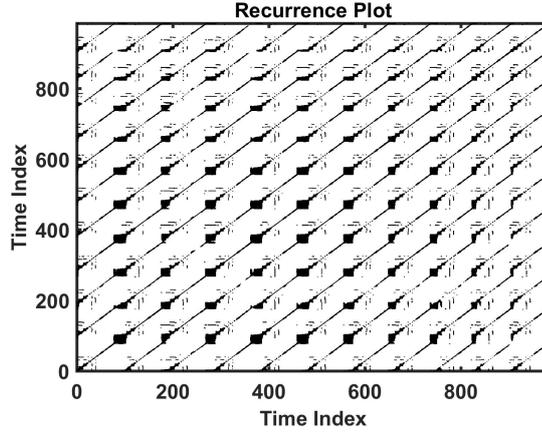


Figure 2.8: RP example for a healthy subject during resting conditions. It is a visualisation of a square matrix in which the matrix elements correspond to those times at which a state of signal recurs.

According to [33], the following values of the parameters introduced above were chosen: $m=10$; $\tau=1$; $r = \sqrt{m} * \text{StdNN}$, with StdNN defined as the standard deviation of the NN series. In the RP, lines are defined as series of diagonally adjacent black points with no white space. The length l of a line is the number of points that the line consists of.

The following measures of RP are computed: recurrence rate (REC) defined in Eq. 2.18; maximal length of lines (l_{\max}); mean length of lines (l_{mean}); the determinism (DET) defined in Eq. 2.19; the Shannon Entropy (ShanEn) defined in Eq. 2.20.

$$REC = \frac{1}{K^2} \sum_{i=1}^K \sum_{j=1}^K M_2(i, j) \quad (2.18)$$

$$DET = \frac{\sum_{l=2}^{l_{\max}} l * N_l}{\sum_{i=1}^K \sum_{j=1}^K M_2(i, j)}, \text{ with } N_l = \text{number of lines of length } l. \quad (2.19)$$

$$ShanEn = \sum_{l=l_{min}}^{l_{max}} n_l \ln n_l, \text{ with } n_l = \text{percentage of } N_l \text{ over all the number lines.} \quad (2.20)$$

The most used non-linear HRV features are shown in Table 2.3 [46–50].

Table 2.3: Non-linear HRV features.

Non-linear Features	Units	Description and Interpretation
SD1, SD2	ms	The standard deviation of the Poincare' plot perpendicular to SD1 and along SD2 the line-of-identity
ApEn	/	Approximate entropy
SampEn	/	Sample entropy
D2	/	Correlation dimension
 DFA:		Detrended fluctuation analysis:
dfa1	/	Short term fluctuation slope
dfa2	/	Long term fluctuation slope
RPA:		Recurrence plot analysis:
RPI_{mean}	Beats	Mean line length
RPI_{max}	Beats	Maximum line length
REC	%	Recurrence rate
RPadet	%	Determinism
ShanEn	/	Shannon entropy
LLE		Largest Lyapunov exponent, used to estimate the chaotic proprieties or sensitivity to the initial conditions of RR intervals dynamics.
LLE (HF)	/	Series filtered in high-frequency band
LLE (LF)	/	Series filtered in low-frequency band

2.2.1.2 Factors influencing heart rate and its variability

Demographic factors (e.g., age, sex and disease history), lifestyle factors (e.g., physical activity, alcohol intake), modifiable risk factors (e.g., hypertension, overweight) and neuropsychological factors (e.g., stress) have all been shown to influence heart rate and its variability. Although it is possible to expect certain differences in baseline (resting) HR and HRV depending on the type of patient or subject, there are multiple factors that contribute to these differences [56]. Therefore, it is difficult to categorise or assess subjects without using demographic data and other factors. The major factors that lead to inter-subjects differences in HR and HRV (independently of the intra-subject factors) are:

Age: age is one of the strongest factors that influences HRV values. Lower HRV generally indicates an increased biological age (older). Higher HRV is correlated with increased fitness, health, and youthfulness [57].

Sex: although age and other factors play a stronger role in influencing HRV than gender, notable gender dependencies of short term HRV indices have been ob-

served. The depression of HRV with age tends to be more marked in males and post-menopausal women. Males typically have lower HRV than females within the same age ranges. This indicates that males exhibit stronger sympathetic tendencies over parasympathetic [57]. The major age-related HRV differences for both genders are between ranges 35 to 44 years and 45 to 54 years. This suggests that the gender-related HRV differences in the younger ages are probably caused by the different hormonal situations leading to a higher sympathetic activity and a lower parasympathetic tone in men and women. In other words, age-dependent and gender-independent changes in HRV indicate diminished parasympathetic activation with increasing age.

Disease history: there are many factors contributing to wellbeing and health status that can markedly affect HRV values. Risk factor conditions and disease onset lower HRV indicating an autonomic imbalance. Medication and other factors can artificially skew HRV values, making them not comparable between individuals. This should be considered when using HRV demographic data. Also, family history should be carefully monitored, as the subjects with a parental history of cardiovascular disease or hypertension could show early symptoms of cardiovascular autonomic impairment [58, 59].

Exercise: exercise immediately induces an increase in HR through a mediated vagal withdrawal. Less fit, aged or extremely youthful hearts have a higher resting HR. Training increases the amount of cardiac muscle and stroke volume to produce a higher maximum and lower resting HR. In fact, high cholesterol levels tend to be associated with lower HRV [57].

Alcohol and drugs: alcohol and drug consumption also have an effect on HRV. Alcohol drinkers and drug users have a low HRV, but this effect is reversible when they stop drinking alcohol or use drugs. It induces an HRV decrease, which could be related to a sympathetic activation or a parasympathetic inhibition [60].

Hypertension: it has been hypothesised that in hypertensive patients, an increased sympathetic and reduced vagal are coupled with an enhancement of vasomotor sympathetic modulation [61]. The severity of hypertension is related to the severity of impairment of cardiac autonomic control measured by time and frequency domain analysis of HRV [62]. HRV values have shown to be lower in hypertensive patients than in normal subjects [63, 64].

Overweight: several studies have documented reduced HRV among overweight and

obese individuals. In fact, obesity provokes a reduction in vagal tone coupled with an increase in cardiac sympathetic activity [65].

Circadian Rhythms: these are defined to be variations in biological activity that appear to have a natural cycle of between 23 and 27 hours, but are often locked into the 24-hour day-night cycle (due to light exposure) [66]. The day-time rhythms influence HRV values, as the circadian rhythm in HRV prepares us for the activities of the day.

Stress: several studies have been conducted to investigate the effect of stress on HRV. A significant change in HRV has been observed, in particular, many stress sources induce low HRV [1].

In the light of this, depressed HRV can be used as a predictor of risk after acute myocardial infarction [67] and as an early warning sign of diabetic neuropathy [68], stress [1] and risk of falls [4].

Moreover, based on these factors, eligibility criteria for the sample population enrolled in the studies presented in Chapter 5 and 6 were carefully selected.

2.3 Theoretical limits of biomedical signal processing and machine learning in real-life settings

The growing interest in monitoring the interaction between the CVS and the ANS via wearable devices has brought attention to some theoretical limitations and challenges (Fig. 2.12) in the processing of acquired biomedical signals and in the development of algorithms for the detection and prediction of adverse healthcare events using data-driven machine learning techniques.

In Fig. 2.9 is shown the most standard and widely used approach to analyse data from wearable sensors in order to perform tasks such as detection, prediction and decision making.

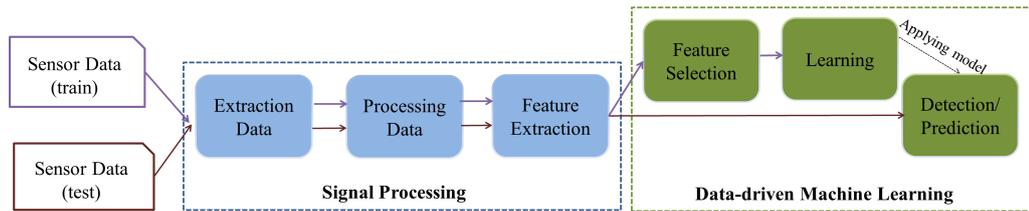


Figure 2.9: Data analysis for wearable sensor data. Adapted from [69]. The main steps of the data mining approach consist of: data extraction, data pre-processing and feature extraction; feature selection and modelling data learning the input features to perform the tasks such as detection and prediction

2.3.1 Biomedical signal processing in real-life settings

Biomedical signal processing involves the analysis of information captured through physiological instruments to provide useful information upon which clinicians can make decisions [70]. Through a review of the existing literature, several theoretical limitations and problems have arisen around signal processing techniques when applied to real-life data. The main identified limitations are the following (Fig. 2.12):

- *Noise and Motion Artefacts.* One of the first limitations regarding physiological signal processing is due to noise and motion artefacts. In fact, as most of the biosensors have an interface to the skin of the subject, an artefact originating from the movement of the body has major implications for the overall system robustness from the quality of the data to the transmission flow rate [71]. As such, artefacts have to be reduced to a minimum and advanced digital signal processing algorithms for noise and artefact reduction should be developed taking into account behavioural signals (i.e., an acceleration signal). In fact, although noise and motion artefacts can be controlled to a high degree in laboratory environments, in real-life settings they represent a big challenge.
- *Context Awareness.* Another important theoretical limit in the processing of data in real-life settings is the difficulty in understanding the context in which the subject is. Human movement monitoring gives information about body gestures and movements but also important information regarding the type of activity the subject is doing, mobility and engagement with their environment. Integrating acceleration signals in wearable devices could bring an advancement in the processing of the signal. In fact, as also suggested in [72], analysis of physiological signals is more meaningful when presented

along with situational context awareness which needs to be embedded into the development of continuous monitoring and predictive systems to ensure its effectiveness and robustness.

- *Multiple Measurements.* Recently, many sensor signals embedded in wearable devices are very different in nature, and the inclusion of other sensing modalities, such as a Global Position System (GPS), audio, camera, or ECG yield diversify sample rates and data characteristics. In addition, data may not necessarily be acquired continuously, but rather (depending on demand) sporadically or with non-uniform sample rates, adding to the heterogeneity and variety in the data. In fact, signal processing methods for dealing with non-uniformly sampled data, specifically data containing large temporal gaps, are not well developed when compared to the methods available for uniformly sampled data. Also, filtering or smoothing non-uniformly sampled longitudinal data to remove noise is not a trivial exercise, and interpolation and re-sampling can lead to false confidence in parameter values where long periods of data are missing [69].
- *Data Visualization.* The integration of multiple measurements brings to light another limit: data visualization. It is quite important for a patient or end-user outfitted with a variety of sensors or with a single wearable device monitoring different signals to be able to read the main information coming from the wearable sensor. However, extracting meaningful information and presenting this information in a format suitable for physicians, patients and final users are non-trivial tasks.
- *Time Horizon.* Another important problem regarding signal processing is the quasi-real time monitoring performed by many wearable sensors embedded in smartwatches and mobile phones, which require an ultra-short recording (below standard recommendations) of the biomedical signals. Nevertheless, many of the biomedical signals lose reliability and accuracy when signal lengths go below the standard recommendations. Therefore, great attention needs to be paid to the processing of ultra-short signal excerpts acquired by wearable sensing devices in real-settings [69].
- *Stationarity.* Most biomedical signals are non-stationary implying that their statistical characteristics do change with time. This is mainly due to the physiological systems generating these signals. As it is well-known, physiological systems are time-varying, non-stationary and non-linear and this makes

the application of analysis tools, usually made for linear, stationary systems analysis, to the study of biomedical signals challenging.

- *Frequency Domain Analysis.* Frequency domain analysis of biomedical signals is a common analysis used to detect non-obvious anomalies. However, as a result of the time horizon problem, many applications do not consider the length of the signals as a limitation for frequency analysis.

2.3.2 Data-mining and machine learning analysis in real-life settings

Data-mining and machine learning analysis are computing processes to discover and learn patterns in small or large datasets. Good use of data-mining and machine learning analysis are the key steps to develop trustworthy algorithms able to predict adverse events and detect the onset of risky well-being situations. Although there are more advanced machine learning techniques- such as deep machine learning-, in this research more traditional machine learning techniques are explored. However, deep learning is a subset of machine learning, and machine learning is a subset of AI, which is an umbrella term for any computer system able to perform tasks that normally require human intelligence (Fig. 2.10) [73].

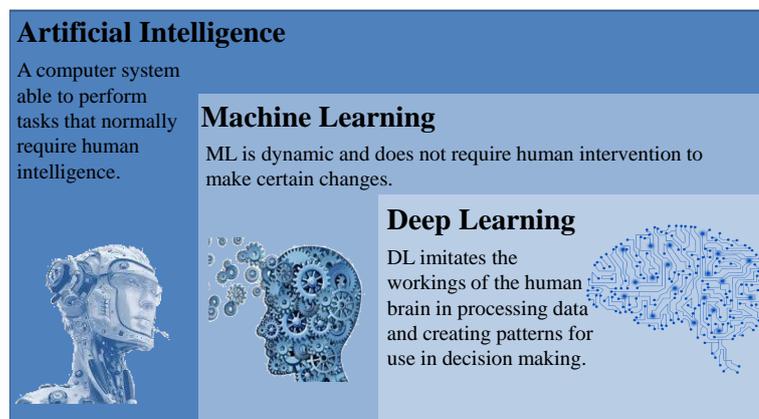


Figure 2.10: Artificial intelligence, machine learning and deep learning.

The choice of using more traditional machine learning lies in the amount of data available in this research. The typical type of input data for traditional machine learning approaches are low volume and structured data [74]. Features extracted from time series or small time series with repeated patterns are generally used. In fact, the most important difference between deep learning and traditional machine learning techniques is their performance as the scale of data increases.

When the datasets are small, deep learning algorithms do not perform as well as traditional machine learning techniques. This is because deep learning algorithms need a large amount of data to understand it perfectly. Moreover, deep learning algorithms heavily depend on high-end machines, contrary to traditional machine learning algorithms, which can work on low-end machines. Furthermore, deep learning algorithms take a long time to train because there are so many parameters in a deep learning algorithm that training them takes longer than usual (Fig. 2.11) [73, 75].

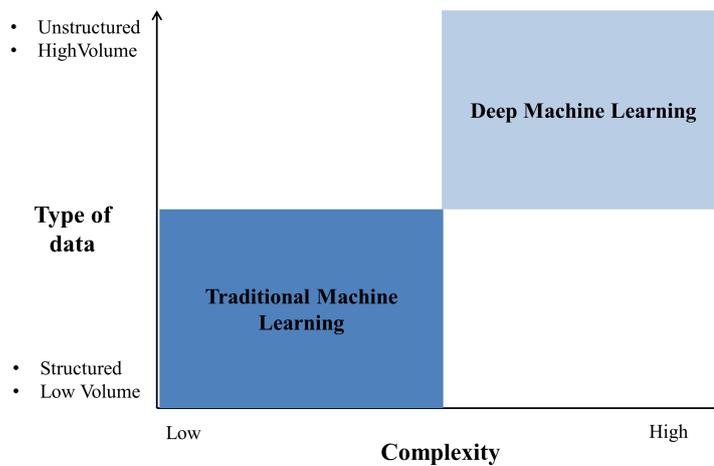


Figure 2.11: Difference between traditional machine learning and deep learning. The graph represents complexity VS type of data for the traditional machine learning and deep learning techniques. Traditional machine learning presents low complexity models and it is mainly used for structured, low volume data; whereas deep learning presents high structural complexity models and it performs better with unstructured, high volume data.

Therefore, in this research more traditional machine learning techniques are used and their main limitations are investigated (Fig. 2.12):

- *Statistical Analysis.* An important phase of data-mining is the use of statistical tools to extract useful information to perform feature selection process and inform the next steps in the development of machine learning models. The nature of the physiological data is extremely broad (continues/discrete or normally distributed/asymmetrically distributed and so on), therefore, great attention needs to be paid to the statistical tools used for each different physiological signal.
- *Feature selection.* An essential part of any classification scheme is feature se-

lection. Features depend on the signals and can be signal variability indices, power densities in physiologically relevant frequency bands, signal model coefficients (i.e., autoregressive or lumped models), transfer and coherence functions, or entropies [71]. The number and choice of attributes are critical to the success of a classifier [76]. The choice of a pertinent small size feature set can improve the personal classifiers' task and machine learning methods, reducing the risk of over-fitting. However, the major limitation lies in how to identify which are the best features and the minimum number of features that can be extracted from the data in order to make a reliable determination of the well-being status of the subject. In fact, as also stated in [76], successful machine learning methods are dependent on the ratio of the number of available training examples of any dataset, big or otherwise, to the number of features extracted. Moreover, for complex or novel data sets, little domain knowledge is available to steer the feature selection process.

- *Model Complexity.* Another problem is model complexity. Many classification algorithms are highly complex which could lead to overfitting problems and they can be computationally expensive. In machine learning, model complexity often refers to the number of features or terms included in a given predictive model, as well as whether the chosen model is linear, nonlinear, and so on. It can also refer to the algorithmic learning complexity or computational complexity. A model with a lower complexity is not only easy to implement in smart devices but will also give lower errors on future real world data.

In the next chapters, model complexity refers to the number of features or terms included in a given predictive model.

- *Rare Events Detection/Prediction.* Automated detection and prediction of rare adverse situations, like a fall, a cardiac event, or some other dangerous situation could lead to overfitting problems [77]. In fact, due to the relative rarity of these events, the algorithms reported in the existing literature show high numbers of false positives [77]. This is a major challenge for scientific researchers who still try to find robust algorithms to predict rare events in real-life settings reducing the number of false positives, and overfitting problems. In fact, frequently the cohort size and monitoring time cannot both be increased to capture a sufficient number of events in order to obtain statistical significance and power [77].
- *Small Datasets.* Small datasets are characteristics of the biomedical engineering domain. In fact, whilst some adverse events are explored on larger

populations, there are many others that are not. This is mainly due to the scarcity of good quality data in real-life settings, the complexity and high cost of experiments, and also due to the miscalculation of sample size in experimental studies, which is mostly computed through statistical methods that may be unsuitable for predictive modelling. Therefore, a big challenge of data-mining and machine learning is to develop robust classifiers using small datasets that will work also on wider samples. As a consequence, the translation of inadequate algorithms mainly developed in-lab settings, hence using small samples, to real-life situations using wearable sensors can give misleading information on the subject's status. In the light of this, it is essential to build a robust algorithm through validation and testing procedures in order to investigate the accuracy and the correctness of the proposed model before being embedded in wearable sensors.

- *Dataset Annotation.* The process of annotating data is expensive and time-consuming. Therefore, to confront this challenge, the efficacy of data-mining and machine learning in an unsupervised context should be investigated using unlabelled datasets.

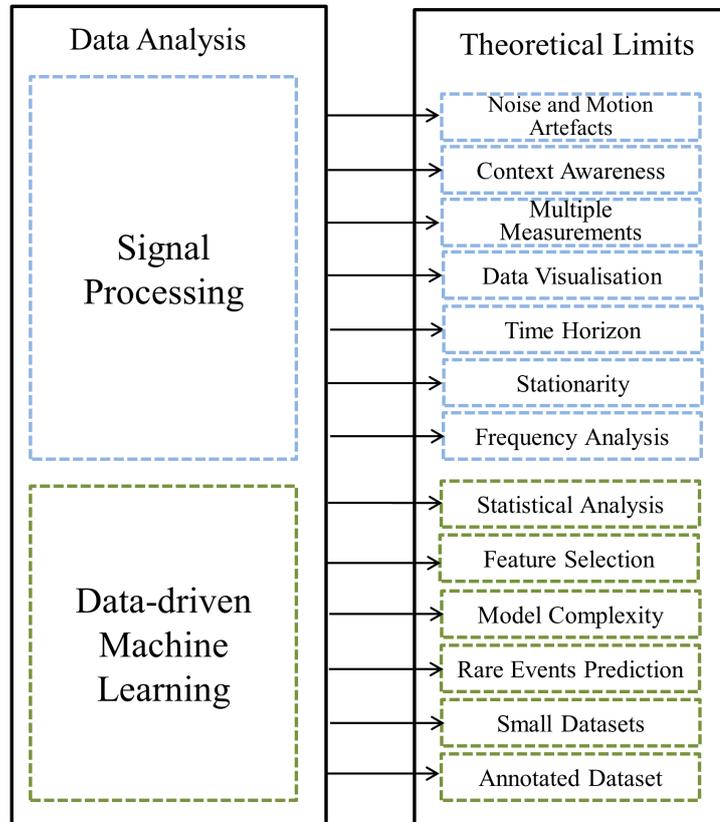


Figure 2.12: Theoretical limitations of applied biomedical signal processing and data-driven machine learning in real-life settings. Theoretical limitations in the monitoring of CVS and ANS in real-life settings via wearable sensors were identified in the two main aspects of data analysis (i.e., signal processing and data-driven machine learning).

2.4 Data mining

Three types of data mining tasks are predominant in the literature. These three tasks are: prediction and anomaly detection, which may include the subtask of raising alarms, and diagnosis where a decision-making process is made to categorise the data into different groups depending on the diseases.

Fig. 2.13 provides an outline of the three most used data mining tasks in relation to the vital signs that can be measured by wearable sensors. In this research, attention is mainly focused on the detection and prediction of adverse healthcare events in real-life settings as highlighted in red in Fig. 2.13. ECG, which provides the most rich data, is predominantly used for all tasks in comparison to the other

types of sensors.

Prediction Prediction is a widely used approach in the data mining field that helps researchers to identify events which have not yet occurred. This approach is gaining more and more interest among the healthcare providers in the medical domain since it helps to prevent further chronic problems and could lead to a decision about prognosis. The role of predictive data mining considering wearable sensors is non-trivial due to the requirement of modelling sequential patterns acquired from vital signs. This approach is also known as supervised learning where it includes feature extraction, training, and testing steps while performing prediction of the data behaviour. Predictive models are currently used to predict accidental falls, strokes, blood glucose levels, mortality prediction and a predictive decision-making system for dialysis patients [69]. For the sake of unexpected situations and conditions in environmental health monitoring (e.g., the home), the difficulty of using predictive models is higher than controlled places such as clinical units. Therefore, there are several challenges in predicting adverse healthcare events using experimental wearable sensor data to perform non-clinical health monitoring (i.e., in real-life settings).

Anomaly detection Anomaly detection is the task of identifying unusual patterns which do not conform to the expected behaviour of the data [69]. The retrieved abnormal patterns in physiological data are significantly valuable in the medical domain. Anomaly detection techniques are often developed based on classification methods to distinguish the data set into normal classes and outliers. Some studies have used domain knowledge and predefined information to detect anomalies for decision making such as anomaly detection in sleep episodes and for finding stress levels.

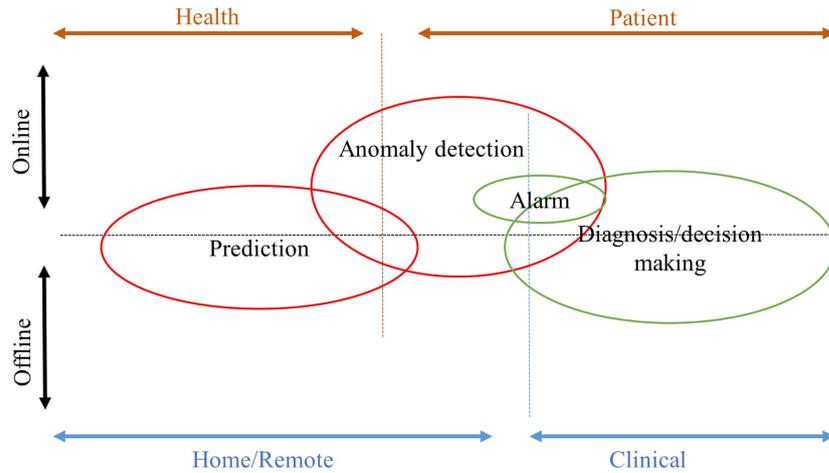


Figure 2.13: Data mining tasks. Adapted from [69]. It provides a depiction of each task in relation to three dimensions. The first dimension involves the setting in which the monitoring occurs (home or clinical settings). A second dimension shows the main data mining tasks in wearable sensors with respect to the type of subjects used (subjects in good health or patients with medical records). The third dimension considers the three main data mining tasks in relation to how the data are processed (online or offline). The tasks explored in this thesis are highlighted in red.

As aforementioned, although there are different methods to accomplish data mining tasks from a combination of deep-learning methods and more traditional machine learning methods, this research is focused on more traditional machine learning methods, which are very effective in predicting patient health status and assessing disease severity.

In machine learning, there are two main approaches, as shown in Fig. 2.14:

- supervised learning, which assumes that training examples are classified (labelled by class labels). The predicted algorithm is based on both input and output data.
- unsupervised learning, which concerns the analysis of unclassified examples. It is used to discover an internal representation from input data only.

In both cases, the goal is to induce an algorithm for the entire dataset or to discover one or more patterns that hold for some part of the dataset [74]. In Fig. 2.14, the most common categories of algorithms used for supervised and unsupervised learning are reported.

In this research, supervised learning, in particular classification algorithms, are explored.

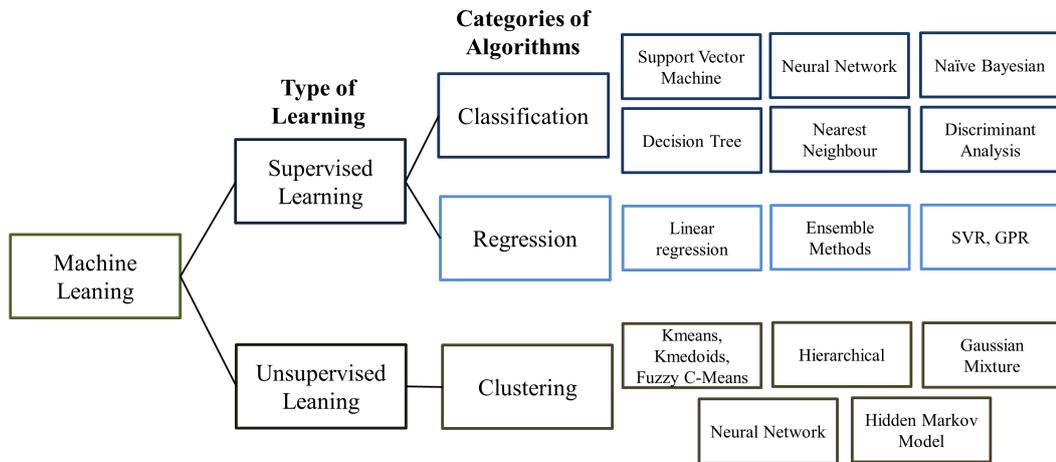


Figure 2.14: Different types of machine learning methods and categories of algorithms. SVR: Support Vector Regression; GPR: Gaussian Process Regression.

Machine learning requires a lot of apriori knowledge and signal pre-processing in order to extract parameters from the acquired time-series. In fact, in any health monitoring system, the main steps of a data mining approach consist of: (Fig. 2.15):

1. signal pre-processing and feature extraction;
2. feature selection;
3. modelling data: learning the input features to perform the tasks such as detection, prediction and decision making.

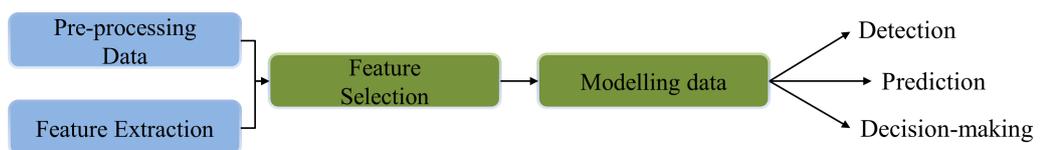


Figure 2.15: Data mining main steps to develop a reliable algorithm for the detection and prediction of an healthcare event as well as for diagnostic decision-making process. In cyan the signal processing steps; in green the data-driven machine learning steps.

2.4.1 Signal pre-processing and features extraction

Signal pre-processing is the first step in any health monitoring device [69]. It mainly consists of:

1. data extraction;
2. signal pre-processing, which includes:
 - filtering unusual data to remove artefacts (i.e., outliers);
 - removing high frequency noise, which is the main source of noise due to electrical activities of other body muscles, baseline shift because of the respiration, poor contact of electrodes, equipment or electronic devices [78];
3. feature extraction.

Having a robust data pre-processing stage requires adequate information about the data themselves. In other words, understanding the type of input data is the prerequisite of any data processing system in order to handle significant issues such as: selecting the proper data mining approach, designing and adjusting new methods and features, and tuning the parameters of the data analysis.

Due to the occurrence of noise, motion artefacts, and sensor errors in any wearable sensor networks, a pre-processing of the raw data is necessary [69].

Regarding ECG signals, the major noise problems are due to [15, 79, 80]:

1. power line interference (50Hz);
2. electrode pop or contact noise, due to the loss of contact between the electrode and the skin manifesting as sharp changes in the ECG signal;
3. electrode motion artefacts, due to the movement of the electrode away from the contact area on the skin, leading to variations in the impedance between the electrode and the skin causing potential variations in the ECG and usually manifesting themselves as rapid (but continuous) baseline jumps or complete saturations for up to 0.5 seconds;
4. electromyographic (EMG) noise, due to muscle contractions lasting around 50ms;
5. baseline drift, usually due to respiration;
6. data collecting device noise, generated by the signal processing hardware, such as signal saturation.

The signal pre-processing flow for an ECG recording is shown in Fig. 2.16.

In the studies presented in the next chapters, the signal pre-processing is carried out with the help of specialised software and manually checked. However, for completeness, the entire process is reported in this section.

The first step consists of data extraction, artefact identification and correction of noise and artefacts. The pre-processing of the ECG usually includes at least bandpass filtering to reduce power line noise, baseline wander, muscle noise, and other interference components [80, 81].

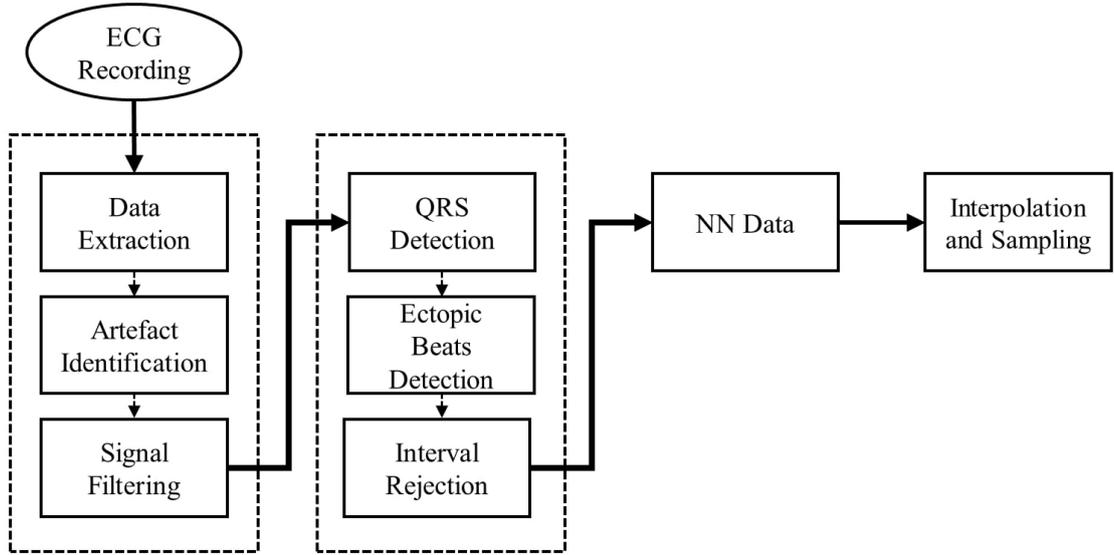


Figure 2.16: Flowchart summarising the individual steps when pre-processing ECG signals to obtain data for HRV analysis.

The signal filtering block is broken down into four separate distinct filtering procedures:

1. 5-15Hz band pass filtering. A low pass filter to remove high-frequency noise (such as 50Hz mains interference) is followed by high-pass filtering to remove low-frequency components due to breathing (at around 1Hz or below).
2. Slope information extraction. Differentiating the signal emphasises changes from the baseline.
3. Squaring. This emphasises the higher frequencies (where the R-peak is to be found) and ensures that all the data are positive for the final stage of filtering.
4. Time averaging. Integrating the squared signal within a moving window gives a measure of how the energy is distributed in the ECG and aids fiducial point localisation.

After pre-processing, the QRS complex is detected in order to derive a meaningful NN tachogram. The inter-beat intervals or NN intervals are obtained as differences

between successive QRS complexes. The accuracy of the location in time of each peak, and hence, the accuracy of the value of each inter-beat interval that comprises the NN tachogram, is dependent on the sampling rate at which the ECG is digitised. A minimum sampling frequency of at least 250Hz is required. In the existing literature, there are many algorithms used for the detection of QRS complex. There are algorithms based upon amplitude and first derivative such as Moriet-Mahoudeaux's methods [82], Fraden and Neuman's scheme [83] and Gustafson's algorithm [84]; others based on first derivatives only, such as Menrads' algorithm [85], The method of Holdinges [86], Balda's method [87] and Ahlstrom and Tompkins' algorithm [88]; other algorithms based on digital filters such as Engelese and Zeelenberg's method [89] and Okada's technique [90].

However, one of the most common algorithms for detecting QRS complexes is the Hamilton and Tompkins algorithm [56]. It is, in fact, the algorithm used in the analysis of the ECG recordings in Chapters 5 and 6.

It applies a set of heuristic rules after carefully removing any noise and artefacts. The following set of heuristics and rules are applied:

1. A peak (of the time-averaged waveform) is located within a segment of the time-averaged wave-form. The segment is defined by noting points where the time averaged waveform exceeds and then falls below a threshold, which is a fraction of the media n-value of the last 10 fiducial points.
2. The fiducial point is then found by a scan-back procedure, searching back through the band-pass filtered data for a peak between the points found in the above step.
3. If the time integrated packet is significantly longer than usual (probably due to dominant P or T waves) then the length of the window of interest is set between 150ms and 250ms.
4. Refractory blanking: as a result of the properties of cardiac tissues, there is a minimum time required to re-polarise. A new peak cannot, therefore, be detected until at least 200ms have elapsed since the last peak detection within the time-averaged signal. If a positive detection occurs within this time frame either the previous or current beat must be false. The algorithm assumes the latter.
5. If a peak is not detected within a certain fraction (slightly greater than unity) of the current average NN interval then a secondary search through the band-filtered data is conducted with lower thresholds on the median filter.

Once the QRS complexes are identified, abnormal beats are rejected. Since beats other than normal (sinus rhythm) beats are generated from outside the normal conduction mechanisms, they are not considered to be representative of autonomic control mechanisms that manifest the observed variability in the NN tachogram. The beat-to-beat intervals that do not correspond to the time differences between two sinus beats must, therefore, be excluded from the NN tachogram [56]. It is common practice to adjust the NN tachogram in order to remove the effect that abnormal (non-sinus, ectopic or aberrant) beats would have on estimating HRV. There are two main arguments for the removal of ectopic beats prior to the calculation of HRV features. Firstly, heart rate modulatory signals involving the brain and cardiovascular system act upon the sinoatrial node (SA). Assessments of autonomic function reflect the ability of this system to stimulate the SA node. Ectopic beats originate from secondary and tertiary pacemakers and this type of locally aberrant beat will temporarily disrupt normal neurocardiac modulation. Secondly, an ectopic beat will often appear late or early with respect to the timing of a sinus beat. This creates a sharp spike in the NN tachogram which is likely to add a significant power contribution to the power spectrum at an artefactual frequency. Since HRV analysis is thought only to be relevant to the timing variations in the sinus rhythm, and the presence of ectopic beats can cause errors in the calculation of HRV metrics a robust method for excluding non-sinus beats and artefacts from the NN tachogram is, therefore, needed [56]. In general, there are two accepted methods for dealing with the effect of ectopic beats in the NN tachogram. If the ectopic or anomalous beats are very occasional, they are removed and interpolation can be used to add a beat where a sinus beat would have been expected to occur. Alternatively, if the incidence of ectopic beats is high within a given segment then it is preferable to eliminate from the analysis, the segments of the HRV signal that contain such a high occurrence. When considering the robustness of HRV analysis to NN interval error, Malik [91] suggests that sequences with a ratio of NN/RR fewer than 90% are excluded. However, he also acknowledge that this an arbitrary threshold. In this research, the ECG recordings were free of ectopic beats and missing data. Therefore, the RR series correspond to the NN series. An illustrative example on how to derive NN (or RR) interval series is shown in Fig. 2.17.

When each QRS complex is detected, the so-called normal-to-normal (NN) intervals (that is all intervals between adjacent QRS complexes resulting from sinus node depolarizations) is determined (2.17) and HRV features can be extracted in time, frequency and non-linear domain as described in section 2.2.1.1. The extracted HRV features can be then input to a data-driven machine learning algorithm to

detect or predict adverse healthcare events. However, different steps are required to develop a trustworthy algorithm as described in the next section.

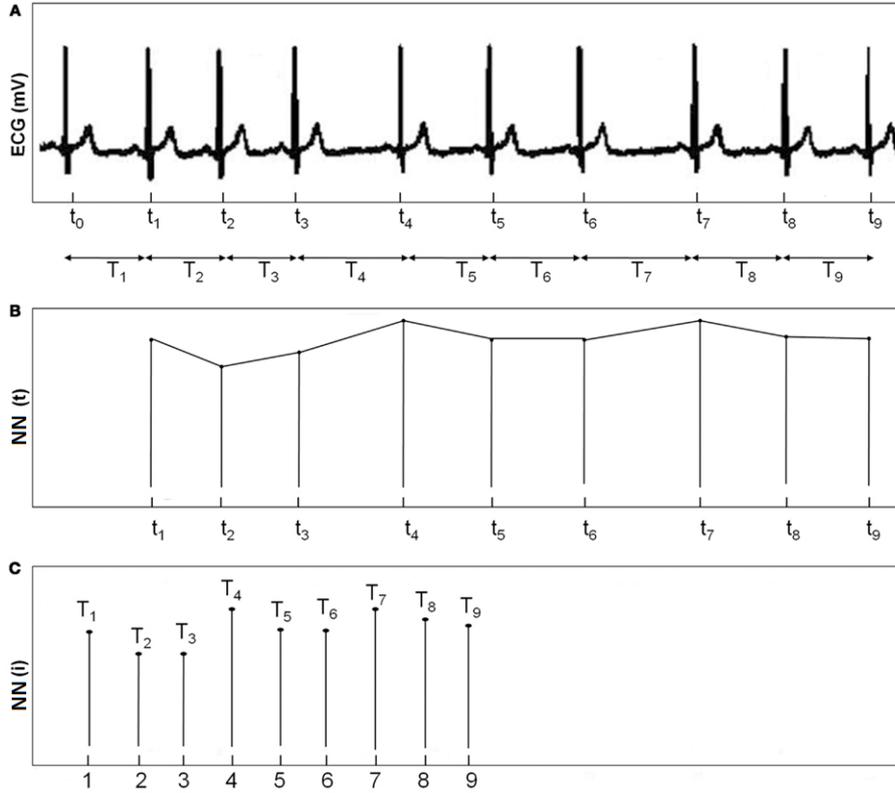


Figure 2.17: An example of NN interval time series. (A) ECG with an event series of R-peaks, which can also be defined as N-peaks due to lack of ectopic beats. (B) Interpolated NN interval time series. (C) NN interval time series [92].

2.4.2 Feature selection and data-driven machine learning techniques

Developing a classifier consists of several steps [76]:

1. choosing a method of analysis;
2. feature selection (i.e., choosing a set of features or attributes that will be used to classify the subjects);
3. training and validating the classifier;
4. testing the classifier;
5. evaluating the classifier and potential errors in the classification.

Each step presents opportunities to introduce bias and error into the process.

Choice of a method of analysis Data mining techniques have progressed significantly in the past few years opening new possibilities for achieving suitable algorithms for wearable sensors. Still, despite these developments, their application to health monitoring is hindered by the limitations (e.g., unbalanced data) that are present in data from wearable sensors which create new challenges for the data mining field. The properties of the dataset and experimental conditions influence the choice of method.

Choice and number of features The number and choice of features are essential to the success of a classifier. Too many features relative to the number of “events” (e.g., sick individuals) leads to overfitting, which results in learning the data instead of discovering the trend that underlies the data. In other words, it is like when one fits data to a high order polynomial: if the order of the polynomial is too large relative to the number of data points, a good fit will be achieved but the polynomial will not capture the trends in the data and have no predictive value for a new set of data points [76]. As a rule of thumb, more than 10 “events” are needed for each feature to result in a classifier with reasonable predictive value [76]. However, many biomedical studies typically involve a small number of subjects, and therefore, there are limited numbers of parameters that can be used in the development of a classifier.

Another important issue is the choice of features. For a diagnostic application, the features must have some bearing on the disease [76]. Therefore, when looking at physiological data, such as an ECG, only the features that are considered to be medically significant to clinical phenomena should be included to describe the signal. By contrast, biomedical engineers often train classifiers using abstract features of a signal, but a classifier would be useful only if the features capture a significant amount of medically relevant information, and they are independent and do not contain confounding variables. For physiological data, using feature selection techniques leads to a reduction in the dimensions of the input sensor data. Three most common approaches for dimension reduction in the medical domain are: Principal Component Analysis (PCA) [93], Independent Component Analysis (ICA) [94], and Linear Discriminant Analysis (LDA), which statistically select the subset of the most significant features. Other tools for feature selection used in the existing literature include threshold-based rules, analysis of variance (ANOVA), Fourier transforms, Wrapper [95] and Relief Attribute Evaluation with Ranker [74]. Although most of the proposed frameworks in the healthcare domain contain feature selection processes, the main challenge is still to balance between the optimal feature selection

methods and their costs for wearable systems.

Training and validation of classifier Training a classifier is the process of taking content that is known to belong to specified classes and creating a classifier on the basis of the unknown content. Several existing studies use statistically justified validation methods such as K-fold or leave-one-out cross-validation (LOO). In these approaches, the dataset is divided in K-folders or one sample, which is removed from the training set, the classifier is recalculated using the remaining training set, and then applied to the holdout samples as a test. This process is repeated in turn for each member of the training set (K). Some machine learning techniques combine training and validation of classifiers in one process. However, the classifier cannot be validated using the same data that are used to develop the classifier in the first place, which would introduce circularity. In fact, “independence” is a theoretical construct that impacts on the external validity of the model.

Testing the classifier Testing a classifier involves testing it on a set of subjects (the testing set) that is independent of the training set. When the dataset is large, one can simply divide it into a training and testing set (hold-out method) or split the dataset into more folders designed for different purposes.

The quality of the biomedical engineering literature on these topics is extremely varied. At the low end of the quality scale, one can find many studies that report no testing at all, but merely show that the classifier works well on the training set, which tells nothing about the predictive value of the classifier when faced with new data. Many other studies lack sufficiently clear description of the validation and testing methods which enables readers to judge the validity of the work.

Evaluating a classifier and potential errors “Evaluation” means estimating the error rate of a classifier. The estimates give an idea about how well the classifier may perform on future unseen cases. After all, its performance on unseen data, instead of known data, is what really matters. Machine learning divides classification onto binary, multiclass, and hierarchical tasks. Evaluation of the performance of a classification model is based on the count of test records correctly and incorrectly predicted by the model. These counts are tabulated in a table known as a confusion matrix. As binary classification problems are explored in this thesis, Table 2.4 shows the general confusion matrix for a binary classification problem.

Table 2.4: The confusion matrix for a binary classification problem.

		True Condition		
		Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive		True Positive (TP)	False Negative (FN)
	Predicted condition negative		False Positive (FP)	True Negative (TN)

The correctness of a classification can be evaluated by computing the number of correctly recognised class examples (true positive), the number of correctly recognised examples that do not belong to the class (true negative) and examples that either are incorrectly assigned to the class (false positive) or that are not recognised as class examples (false negative) [96]. Table 2.5 presents the most often used measures for binary classification based on the value of the confusion matrix [96].

Table 2.5: Measures for binary classification using the nation of Table 2.4.

Measure	Formula	Evaluation Focus
Accuracy (ACC)	$TP + TN / TP + FN + FP + TN$	Overall effectiveness of a classifier
Precision (PRE)	$TP / TP + FN$	Class agreement of data labels with positive labels given by the classifier
Sensitivity (SEN)	$TPR = TP / TP + FN$	Effectiveness of a classifier to identify positive labels
Specificity (SPE)	$TNR = TN / FP + TN$	How effectively a classifier identifies negative labels
Area Under the Curve (AUC)	$\frac{1}{2}((TP/TP + FN) + (TN/TN + FP))$	Classifier ability to avoid false classification
LR+	TPR / FPR	Positive Likelihood Ratio
LR-	FNR / TNR	Negative Likelihood Ratio
Diagnostic Odds Ratio (DOR)	$LR + / LR -$	It is a measure of the effectiveness of a diagnostic test

TPR: True Positive Rate, FPR: False Positive Rate; FNR: False Negative Rate, TNR: True Negative Rate.

A useful visual tool to evaluate any classifier error is the Receiver Operating Characteristic (ROC) curve, which is the plot of sensitivity versus 1-specificity [97]. A ROC curve is a technique for visualising, organising and selecting classifiers based on their performance. ROC curves have long been used in machine learning methods to depict the trade-off between hit rates and false rates of classifiers [98]. ROC analysis has been extended used for visualising and analysing the behaviour of diagnostic systems [99]. The medical decision making community has an extensive literature on the use of ROC curves for diagnostic testing [97]. In other words, this curve plays a central role in evaluating diagnostic ability of tests to discriminate

the true state of subjects, finding the optimal cut off values, and comparing two alternative diagnostic tasks when each task is performed on the same subject [100]. A typical example of a ROC curve is shown in Fig. 2.18.

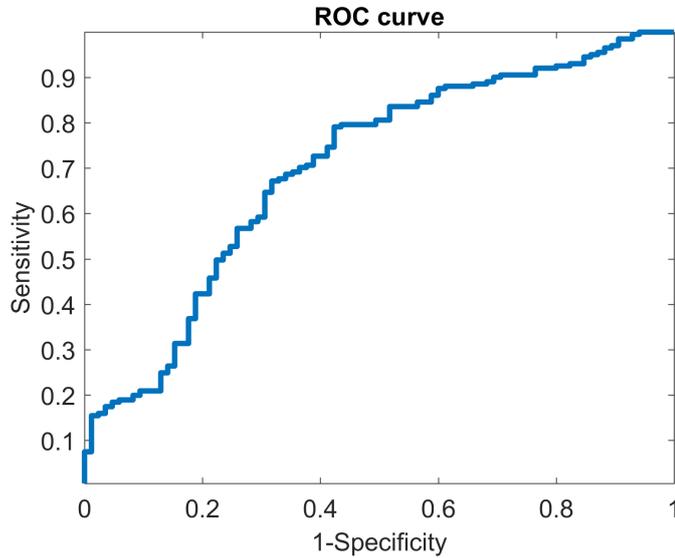


Figure 2.18: An example of ROC curve from one of the studies.

The area under the ROC curve (AUC) is a very widely used measure of performance for classification and diagnostic rules [101]. In fact, AUC is an effective measure of accuracy [100].

2.4.2.1 Common machine learning methods

In this section, the most common machine learning methods used in this research are outlined. Each method is described in technical detail with the most representative examples of how the algorithm has been applied in healthcare services. In addition, the usability, efficiency and challenges in applying each technique in the medical domain are indicated where possible.

Support Vector Machine Support Vector Machine (SVM) is one of the main statistical learning methods which is able to classify unseen information by deriving selected features and constructing a high dimensional hyperplane to separate the data points into two classes in order to make a decision model [102]. SVM methodology is very popular for mining physiological data, not only due to its ability to handle high dimensional data using a minimal training set of features. The SVM method consists of different kernels such as Radial Basis Function (RBF),

polynomial, and sigmoidal, which often are combined to model the input data from multi-sensor features. According to the existing literature, many researches have shown that the SVM method with a polynomial kernel leads to better results than other kernels.

Common health parameters considered by SVM methods include ECG and HR. In fact, SVM has been widely used in medical applications, for example, Hu *et al.* [103] used SVM to find out arrhythmia in ECG signals. They applied a binary classifier version of SVM to categorise ECG signals into normal and arrhythmia classes. Similarly, other studies developed an SVM method for detecting the arrhythmia and seizure episodes using ECG signals. In fact, SVM techniques are often used for anomaly detection and decision making tasks in the healthcare services.

The performance of the method is evaluated with the sensitivity, specificity, and accuracy of the results, according to the binary measures.

Artificial Neural Networks Artificial Neural Networks (ANNs) are artificial intelligence approaches which are widely used for classification and prediction [104]. The ANNs train data by learning the known classification of the records and comparing with predicted classes of the records in order to modify the network weights for the next iterations of learning [69]. Nowadays ANNs are the most popular machine learning method used in the medical domain due to its admissible predictive performance [69]. The prowess of the ANNs is to model highly non-linear systems such as physiological data and where the correlation of the input features is not easily detectable. ANNs have also been applied for multi-sensor networks to help handling the analysis of multivariate data. A version of the ANNs is the Multilayer Perceptron (MLP), which has common been also widely applied in medical applications. This network puts several individual signal quality metrics as input and then optimises the number of nodes and hidden layers in validation iterations. For evaluation of the NN, it employs some common measures as reported in Table 2.5.

There are other variations of ANNs such as Replicator Neural Network (RNN), which are generally used for anomaly and outlier detection. In fact, a recent study conducted by Vu *et al.* [105] proposed a framework to recognise HRV patterns using ECG and accelerometer sensors.

Overall, ANNs result to be not considered as a portable technique to easily apply for diverse data sets.

Decision Trees The decision tree method is one of the most significant learning techniques which provides an accurate discrimination for selected features and an

efficient representation of rule classification [106]. In this method, the most robust features have been detected among the selected features for initial splitting of the input data by creating a tree-like model. There are numerous variations of decision trees, however, the most used in the medical domain is the C4.5, which deals with complex and noisy data. The C4.5 algorithm estimates the error rate of initial nodes and prunes the tree to make a more efficient sub-tree [106]. Another common decision support system is Random Forest (RF) classification. For the construction of each tree of the forest, a new subset of the features is usually picked. For selecting the best tree, the method uses threshold-based rules.

Generally, decision tree methods are limited to the space of the constructed features as the inputs of the model. So, finding hidden information out of constricted features would not be recognisable. Furthermore, since the number of features can impact on the efficiency of the method, decision tree models are not usually applied to big and complex physiological data. In fact, decision trees algorithms have been mainly used in medical applications especially with short term data and when few subjects are available [69].

Naïve Bayes Classifier The Naïve Bayes (NB) classifier greatly simplifies learning by assuming that features are independent given the class. The knowledge generated and used by the NB classifier is simply a table of prior and conditional probabilities approximated with relative frequencies from the training set.

Although independence is generally a poor assumption, in practice a NB classifier often competes well with more sophisticated classifiers [107]. NB has been reliably used in many practical applications including text classification, medical diagnosis, and systems performance management. In fact, the NB method has been lately used in medical settings to diagnose the location of primary tumours, prognosis the recurrence of breast cancer diagnosing thyroid diseases and rheumatology, in which NB performances outperformed other learning techniques. The major advantage of NB is the reliability of the approximation of probabilities [108].

There are different variations of NB such as Multinomial Naïve Bayes (MNB), which is based on Bayes' theorem (Bayes' rule), with the additional incorporation of frequency information and a multinomial distribution for each of the features [109], and the NB network, which removes the bias introduced by the independence assumptions embedded in the NB classifier [110].

Linear Discriminant Analysis Traditional statistical classification methods (e.g., Fisher's Linear Discriminat Analysis (LDA) and Logistic Regression (LR)) have

been extensively used in medical classification problems [111]. The oldest classifier still in use was devised almost 100 years ago by Sir Ronald Fisher. Another variation of LDA is Quadratic Discriminant Analysis (QDA), which uses a quadratic discriminant function. The principle behind both LDA and QDA is that a subject is classified into the group for which its classification function score is higher. Discriminant analysis is widely used also in the field of pattern recognition, which is concerned mainly with images. Although it is currently used in medical settings, its ability is limited due to the randomness of many physiological data acquired by wearable devices.

K-Nearest-Neighbour Classifier IBK is a different kind of search algorithm that can be used to speed up the task of finding the nearest neighbours [112]. IBK uses multi-dimensional feature space in which each dimension corresponds to a different feature. The feature space is first populated with all training data points, each of which corresponds to a particular class. Unknown windows of sensor data are represented in the feature space and the IBK classifier is identified using training data. The value of K typically varies from 1 to a small percentage of the training data and is selected using trial and error, or ideally using a cross-validation procedure. A linear search is the default option but further options include KD trees, ball trees, and so called cover trees. The distance function used is a parameter of the search method [113].

IBK has been widely used in medical settings to differentiate between everyday activities, using different biomedical signals such as HRV, accelerometers and skin conductance.

Overall, the data mining methods described above are mainly used in controlled conditions and clear data sets, but the efficiency of these is often not tested in real-life experiments for healthcare applications.

2.5 Conclusions

This chapter informed on the main methodologies employed regarding biomedical signal pre-processing and machine learning techniques explored in this research. The interaction between the CVS and the ANS was briefly discussed as it opens new scenarios for predicting adverse events and detecting the onset of unhealthy situations in real-life settings. The monitoring of CVS and ANS dysfunction is mainly measured via HRV, which is one of the best known and non-invasive biomedical signals. Therefore, HRV was widely explored in this thesis and analysed in the time,

frequency and non-linear domains, as described in section 2.2.1.1. Moreover, different demographic, life style and neuropsychological factors influencing HRV were carefully analysed in the following study designs, as they were considered in the eligibility criteria of the sample population enrolled in the experiments presented in Chapters 5 and 6.

The main theoretical limits regarding signal processing and data-driven machine learning techniques used to monitor CVS and ANS response to detect and predict adverse healthcare events in real-settings were also discussed. Some of those limitations (i.e., time horizon, feature selection process, rare events prediction, handling small datasets) are then explored and investigated in more details in Chapter 4, providing novel approaches to overcome them.

More traditional biomedical signal pre-processing and machine learning techniques were briefly discussed in sections 2.4.1 and 2.4.2 respectively. In particular, signal pre-processing techniques applied to ECG signals were reviewed and traditional machine learning techniques, including feature selection process, were discussed for binary supervised classifiers used to predict and detect adverse healthcare events.

More details and applications of these techniques are presented in the next chapters.

Chapter 3

Literature Review on Acute Mental Stress and Falls in Later-life

3.1 Chapter overview

The previous chapter introduced the main notions regarding ANS and CVS interaction; the relevant biomedical signal - HRV- used to monitor the ANS and CVS relationship; signal processing and machine learning techniques employed in this thesis and their limitations when applied in real-life settings to predict or detect adverse healthcare events.

This chapter presents the state-of-the-art for the two case studies: mental stress detection and fall prediction in later-life via HRV. Relevant gaps for both case studies are identified.

In particular, the first part of this chapter (section 3.2) aims to provide information on the existing study designs to detect mental stress, HRV features and their pivot values during stress, and the current methods used to assess the validity of ultra-short HRV analysis (deliverables 1a and 1b).

The second part of this chapter (section 3.3) comprises a brief overview of the main fall risk factors, a brief description of prevention and prediction programmes, and a discussion on the existing monitoring technologies used to detect and predict falls in older people. The existing literature on HRV and risk of falling is also explored (deliverable 2a).

3.2 Literature review on acute mental stress detection via HRV

A systematic review of the literature was carried out to understand the relationship between HRV and ANS during stress, to extract significant HRV features, their pivotal values and to inform on future study designs (deliverable 1a). As a consequence of the systematic review, since few studies have investigated mental stress using ultra-short HRV features and even fewer studies have assessed the validity of the latter, a review of the existing methods to assess ultra-short HRV features as good surrogates of short HRV ones was also carried out (deliverable 1b).

3.2.1 Stress definition

Stress is defined by the American Psychological Association as “*the pattern of specific and non-specific responses an organism makes to stimulus events that disturb its equilibrium and tax or exceed its ability to cope*” [114]. In particular, mental stress is defined by Lazarus and Folkman as a form of stress that occurs because of how events in one's external or internal environment are perceived, resulting in the psychological experience of distress and anxiety [115].

Mental stress has been investigated in various fields due to its destructive effects on a daily routine. Stress may cause cognitive dysfunctions, cardiovascular diseases, depression and may lead to illness and death [116]. There is, in fact, an average of 50% of employees, who suffer from work stress [117]. Moreover, stress reduces performance in daily life and particularly in the workplace. This may be particularly relevant for those jobs that expose workers or other persons to risky situations. This is the case of surgeons performing long surgeries, in which the loss of attention or concentration may cause severe effects on the patient, or pilots flying over long distances, whose stress may be dangerous for them and the passengers [118].

Mental stress influences the ANS, which controls our capability to react to external stimuli [115]. Therefore, the acute stress may be evaluated with non-invasive biomarker measurements, which are considered reliable estimators of ANS status. This is the case of HRV, which is considered a reliable means to indirectly observe ANS, also in real-life settings. As also stated in Chapter 2, section 2.2.1, HRV refers to the variations of both instantaneous heart rate and the series of inter-times between consecutive peaks of the R-wave of the ECG (NN series) [15]. This variation is under the control of the ANS, which through the parasympathetic and the sympathetic branches, is responsible for adjusting the HRV in response to external or internal physical or emotional stimuli. A normal subject shows a good

degree of variation of the heart rate, reflecting a good capability to react to those stimuli.

In the recent years, several studies investigated how HRV could be used to assess acute or chronic mental stress in healthy subjects or specific groups of patients of different ages. These studies collected and analysed HRV with different methods and tools: long (24h) or short term (5 min) registrations were utilised, very few studies have used ultra-short term HRV analysis (less than 5 min). HRV features have been extracted from ECG or SpO₂ or via other means with a wide range of devices (i.e., from commercial smartphones to CE marked medical devices for ECG monitoring, including some advanced biomedical amplifiers). HRV has been analysed with commercial software tools, sometimes well-validated ones. HRV features in time and/or frequency have been extracted and both linear and/or non-linear domains have been explored and different statistical tests, pattern recognition or data mining methods, have been used to explore the results [15, 117].

3.2.2 Systematic literature review with meta-analysis

The main aim of this systematic literature review was to revise homogeneously designed studies to provide reliable information about the trends and pivotal values of HRV features during mental stress and inform future study designs.

Therefore, this review systematically investigated studies published in peer-reviewed journals and focused on the associations between acute mental stress and HRV, using short-term and ultra-short HRV excerpts, extracted from ECGs, in healthy subjects above 18 years old.

3.2.2.1 Methods and materials

3.2.2.2 Search strategy

Relevant studies on the detection of acute mental stress through HRV analysis were identified and selected by searching on the PubMed and OvidSP databases. Pertinent articles were searched using Boolean combinations of the following keywords or their equivalent Medical Subject Heading (MeSH) terms: “Heart Rate Variability”, “HRV”, “mental stress”, “psychological stress” and “emotional assessment”. The following criteria were utilised to limit the research: papers published in the last 13 years (since 2004), studies on adult humans (i.e., not animals), not children, not cancer, not pregnancy.

3.2.2.3 Inclusion and exclusion criteria for paper selection

After a first screening of the titles and abstracts, studies were considered suitable for this review if they met all of the following criteria:

- paper published in scientific journals with peer review;
- paper focused mainly on mental stress investigation using HRV;
- the experiments were conducted with a robust design, well described, including repeated measures in the same group of healthy subjects at rest and during stress sessions. Studies were considered robust if the acquisition of HRV was adequately standardised (i.e., stress and rest sessions at the moment of the day to minimise the circadian effect and with the subjects in the same position);
- the study was not focused on chronic stress, and only acute stressors were used, as defined in [119];
- the subjects were humans beings over 18 years old.

Studies were excluded if they:

- focused on chronic stress;
- utilised HRV analysis on excerpts that were longer than 10 minutes;
- enrolled professional athletes and observed them during sport training sessions;
- investigated pain perception;
- reported HRV features of insufficient quality (i.e., features' units not stated, inappropriate statistical descriptors);
- did not report inclusion/exclusion criteria for sample selection.

3.2.2.4 Paper short-listing, data extraction and outcomes of interest

Following the research strategy described above, all the titles responding to the chosen keywords were identified. After excluding repetitions (titles indexed both in PubMed and OvidSp), studies were shortlisted according to inclusion/exclusion criteria, investigating titles, abstracts and full-papers.

All of the shortlisted studies followed the standard recommendations for ECG equipment and commercial software used to assess HRV analysis [15]. The ECG equipment used in the shortlisted studies satisfied the current industrial standards

in terms of signal/noise ratio, common mode rejection and bandwidth [15]. The sampling frequency of the ECG equipment ranged from 250 Hz to 500 Hz, which is considered to be the optimal range [15]. Commercial software used to assess HRV analysis in the shortlisted studies satisfied the technical requirements listed in the recommendations of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology guidelines [15]. HRV features in the time and frequency domains, both linear and non-linear, were extracted from the papers included in this review [45, 116, 120–129]. The changes in HRV features during stress and rest sessions were investigated observing their trends. A trend was represented with arrows pointing up, if the mean value of an HRV feature was increasing during the stress session, with respect to the value of the same feature registered during the resting session. An arrow pointed down was used if the mean value of the HRV feature was decreasing. Two arrows were utilised for significant changes (p -values <0.05). Since papers reported only parameters for feature distributions (i.e., means and standard deviations) the p -values were calculated with a parametric test (i.e., Student's t -test) and not with non-parametric tests, which in some cases would have been more appropriate.

3.2.2.5 HRV features

Regarding linear HRV features in the time and frequency domains, the recommendations of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology guidelines [15] were followed. Therefore, the power spectrum density features were considered if reported in normalized units or ms^2 , for the low-frequency bands (LF, 0.04-0.15 Hz) and the high-frequency bands (HF, 0.15-0.4 Hz). When papers shortlisted for meta-analysis reported features in a non-conventional way, these features were excluded from the pooling or converted into ms^2 , if possible. For instance, three studies [125, 126, 128] reported frequency domain features log-transformed (LF_{log} and HF_{log}). These were untransformed according to [130], using the following formulae:

$$\text{mean } LF_{ms^2} = \exp(\text{mean } LF_{log}) \quad (3.1)$$

$$\text{std } LF_{ms^2} = \text{std } LF_{log} * \exp(\text{mean } LF_{log}^2) \quad (3.2)$$

One study reported frequency features in $\frac{ms^2}{Hz}$ [123] and therefore its results were not reported in this review.

Moreover, according to [15], HRV frequency features (i.e., LF, HF, LF/HF) calculated in excerpts below 2 minutes were excluded from the meta-analysis. In fact, it is generally recommended that spectral analyses are performed on recordings lasting for at least 10 times more than the slower significant signal oscillation period.

3.2.2.6 Statistical analysis and software tools

HRV features were pooled if reported in more than one paper. Standard methods for systematic reviews with meta-analyses were employed in this study to pool the HRV features [131]: mean difference (MD) with 95% confidence intervals (95%CI) and p-values (p).

When the studies report means and standard deviations, the preferred effect size is usually the raw mean difference (MD). In fact, when the outcome is reported on a meaningful scale and all studies in the analysis use the same scale, the meta-analysis can be performed directly on the raw difference in means (henceforth, raw mean difference). The primary advantage of the raw mean difference is that it is intuitively meaningful. Consider a study that reports means for two groups (treated and control), let μ_1 and μ_2 be the true (population) means of the two groups. The population mean difference is defined as reported in Eq. 3.3

$$\Delta(MD) = \mu_2 - \mu_1 \quad (3.3)$$

After estimating the MD, the variance and the pooled standard deviation are reported in Eqs. 3.4 and 3.5 respectively. Let SD_1 and SD_2 be the sample standard deviations of the two groups, and n_1 and n_2 be the sample sizes in the two groups. Assuming that the two population standard deviations are the same so $\sigma_1 = \sigma_2 = \sigma$, the variance of the MD is defined as reported in Eq. 3.4.

$$V_{MD} = \frac{n_1 + n_2}{n_1 * n_2} SD_{pooled}^2 \quad (3.4)$$

where,

$$SD_{pooled} = \sqrt{\frac{(n_1 - 1)SD_1^2 + (n_2 - 1)SD_2^2}{n_1 + n_2 - 2}} \quad (3.5)$$

The confidence interval is then calculated as reported in Eq. 3.6.

$$\begin{aligned} LowCI &= MD - 1.96 * \sqrt{V_{MD}}, \\ HighCI &= MD + 1.96 * \sqrt{V_{MD}} \end{aligned} \quad (3.6)$$

Random or fixed effects models were used according to the studies' heterogeneity measured with a Q statistic test. A significant Q statistic (Eq. 3.7) is indicative of dissimilar effect sizes across studies and to complement the Q test, the I² statistic was also calculated, which provides an index of the degree of heterogeneity across studies. In particular, I² indicates the percentage of the total variability in effect sizes due to between-studies variability, and not due to sampling error within studies [132]. Percentages of around 25% (I² = 25), 50% (I² = 50), and 75% (I² = 75) may be interpreted as low, medium, and high heterogeneity, respectively [132]. The Q statistics is given by:

$$Q = \sum_{i=1}^n \frac{(MD_i - \bar{\theta})^2}{W_i^F} \quad (3.7)$$

where,

$$\bar{\theta} = \frac{\sum_{i=1}^n W_i^F * MD_i}{\sum_{i=1}^n W_i^F} \quad (3.8)$$

where, W^F is defined in Eq. 3.9. The weight for each study was then computed according to the random or fixed effects models. For a fixed model, the weight (W^F) was calculated as reported in Eq. 3.9.

$$W_i^F = \frac{1}{V_{MD}}, \text{ where, } i = 1, \dots, n, \text{ with } n \text{ equal to the number of studies.} \quad (3.9)$$

In the case where the model is random, the weight (W^R) is calculated as reported in Eq. 3.10

$$W_i^R = \frac{1}{V_{MD} + \tau_2}, \text{ where, } i = 1, \dots, n \quad (3.10)$$

$$\tau_2 = \frac{Q - (n - 1)}{U} \quad (3.11)$$

Here Q and U are computed as reported in Eqs. 3.7 and 3.12 respectively, namely

$$U = \sum_{i=1}^n W_i^F - \frac{\sum_{i=1}^n (W_i^F)^2}{\sum_{i=1}^n W_i^F} \quad (3.12)$$

The weight's percentage for both fixed and random effect is calculated as reported in Eq. 3.13.

$$\%W_i = 100 * \frac{W_i}{\sum_{i=1}^n W_i}, \text{ where } n \text{ is the number of studies.} \quad (3.13)$$

Differences and 95% Confidence Interval (CI) were considered significant if the p-

value was less than 0.05.

A Software tool was developed to compute these statistics as reported in Appendix A, section A.1.

3.2.2.7 Matlab tool for meta-analysis

A Matlab tool was developed by the author to support researchers in the extraction of the main parameters of a meta-analysis study according to the input data (Fig. 3.1). Although several software tools for meta-analysis exist, also free, this is the first tool developed in Matlab, which is one of the widely used tools among biomedical engineers. The tool is targeted for researchers having familiarity with Matlab.

The tool is developed to conduct meta-analysis on continuous variables (not dichotomous variables) and is based on the formulae reported in the section above (section 3.2.2.6).

```

Input matrix
Calculate weights
Model Selection calculate Q statistics
  if [Fixed model]
    then calculate weights according to fixed model formulae
  else [Random model]
    then calculate weights according to random model formulae
  end
Calculate individual and pooled statistics
Generate Forest plots

```

Figure 3.1: Pseudocode for the meta-analysis tool.

The main function of the tool is designed to take as input a matrix (Fig. 3.2), each row is expected to be organised as reported in Eq. 3.14:

$$ID_i, N_i^T, O_i^T, SD_i^T, N_i^C, O_i^C, SD_i^C \quad (3.14)$$

where n is the number of studies included in the meta-analysis and:

ID_i is the row containing the studies' ID (text);

$N_i^{T(C)}$ is the number of subjects enrolled in the treatment (control) group in the study i ;

$O_i^{T(C)}$ is the outcome observed in the treatment (control) group in the study i ;

$SD_i^{T(C)}$ is the standard deviation of the outcome observed in the treatment (control) group in the study i .

Moreover, the first row should contain the header names.

	1 IDs	2 N_Treatment	3 Mean_Treatment	4 SD_Treatment	5 N_Control	6 Mean_Control	7 SD_Control
1	'Schubert, 2009'	50	33.2000	23.8300	50	96	86.6400
2	'Taelman, 2011'	43	46.7300	19.4800	43	35.4000	16.3500
3	'Tharion, 2009'	18	74.2000	25.9280	18	52.4000	21.6700
4	'Visnovcova, 2014'	70	56.2300	21.6700	70	48.9800	17.9000
5							

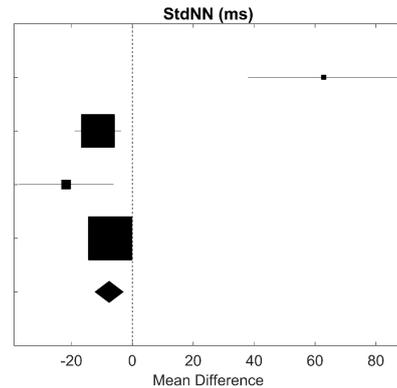
Figure 3.2: An example of input matrix for the meta-analysis Matlab tool. The first column (IDs) reports the first author and year of publication for each study included in the meta-analysis. The second and fifth columns (N_Treatment and N_Control) report the number of subjects enrolled in the treatment and control groups. The third and sixth columns (Mean_Treatment and Mean_Control) report the mean values of outcome of interest. The fourth and the seventh columns (SD_Treatment and SD_Control) report the standard deviation of the outcome observed in the treatment and control groups.

The output of the tool is given as one plot (a forest plot) and one numerical row reporting: the total population, heterogeneity (I^2) and related p-value, model (i.e., random or fixed), the effect size, 95%Confidence Interval (CI) and the relative p-value (Fig. 3.3).

	1 Outcome	2 Tot_Population	3 Isq	4 p_value_Isq	5 Model	6 MD	7 Low_CI	8 High_CI	9 p_value_CI	10
1	'StdNN'	181	91.3806	1.0000e-05	"Random"	1.4874	-16.9392	19.9139	1.0000e-05	
2										

(a) Numerical row reporting the pooled results: the outcome name (StdNN), the total population (Tot_Population), heterogeneity (Isq) and related p-value (p_value_Isq), model (i.e., random or fixed), the effect size (MD), the CI (High_CI, Low_CI) and the relative p-value (p_value_CI).

Study	MD	Low	High	p-val
Schubert, 2009	62.800	37.893	87.707	0.000
Taelman, 2011	-11.330	-18.932	-3.728	0.004
Tharion, 2009	-21.800	-37.411	-6.189	0.010
Visnovcova, 2014	-7.250	-13.834	-0.666	0.033
Pooled	-7.627	-12.286	-2.969	0.001



(b) StdNN Forest plot example.

Figure 3.3: An example of output for the meta-analysis Matlab tool.

According to [131], different formulae are used to calculate the effect size and its 95%CI depending on: the design of the studies pooled (i.e., randomised or not) and the level of heterogeneity which affects the way the weights are calculated. A peculiar feature of the software is the automatic selection of the pooling model. This is delegated to a sub-function that receives as input the matrix described in Eq. 3.14 and returns 0 or 1 according to the fact that there is or not a significant heterogeneity, and therefore, the pool requires or does not require the use of the random model, based on the p-value obtained with a Chi-square statistical test using a conventional level of statistical significance (p-value= 0.05).

The Matlab code was checked for errors and standards compliance through debugging functions. It was also tested to refine the requirements until the design was fully functional and no unintended behaviours were encountered. The precision of the values obtained with the developed Matlab tool was estimated comparing its results with those from the Open-Meta[Analyst][133], which is a widely used and validated software tool.

The main function takes as input the matrix specified in Eq. 3.14 and returns the Forest plots along with the main descriptive statistics. The Unified Modelling Language (UML) sequence diagram for the tool is represented in Fig. 3.4.

The main function (“CallMetaAnalysis”) calls:

1. “ModelSelection” function, which calculates the level of heterogeneity and automatically selects the pooling model. The internal functions are:
 - (a) “Qtest” function, which calculates Q statistics and returns the p-value, based on the “WeightCalculation” function
 - i. “WeightCalculation” function calculates the weights of the studies, based on the fixed model (‘Individual Fixed Weights’)
2. “WeightCalculation” function, which is called again to compute the weights based on the output of the “ModelSelection” function (i.e., fixed or random models). Moreover, it also computes the study's individual effect size and p-value (‘Individual Studies Stat’).
3. “MainStat” function, which computes the main statistics for the pooled value: pooled effect size, its 95%CI, the total population and pooled p-value (‘Pooled Stat’).
4. “Forest” function, which produces the Forest plots.

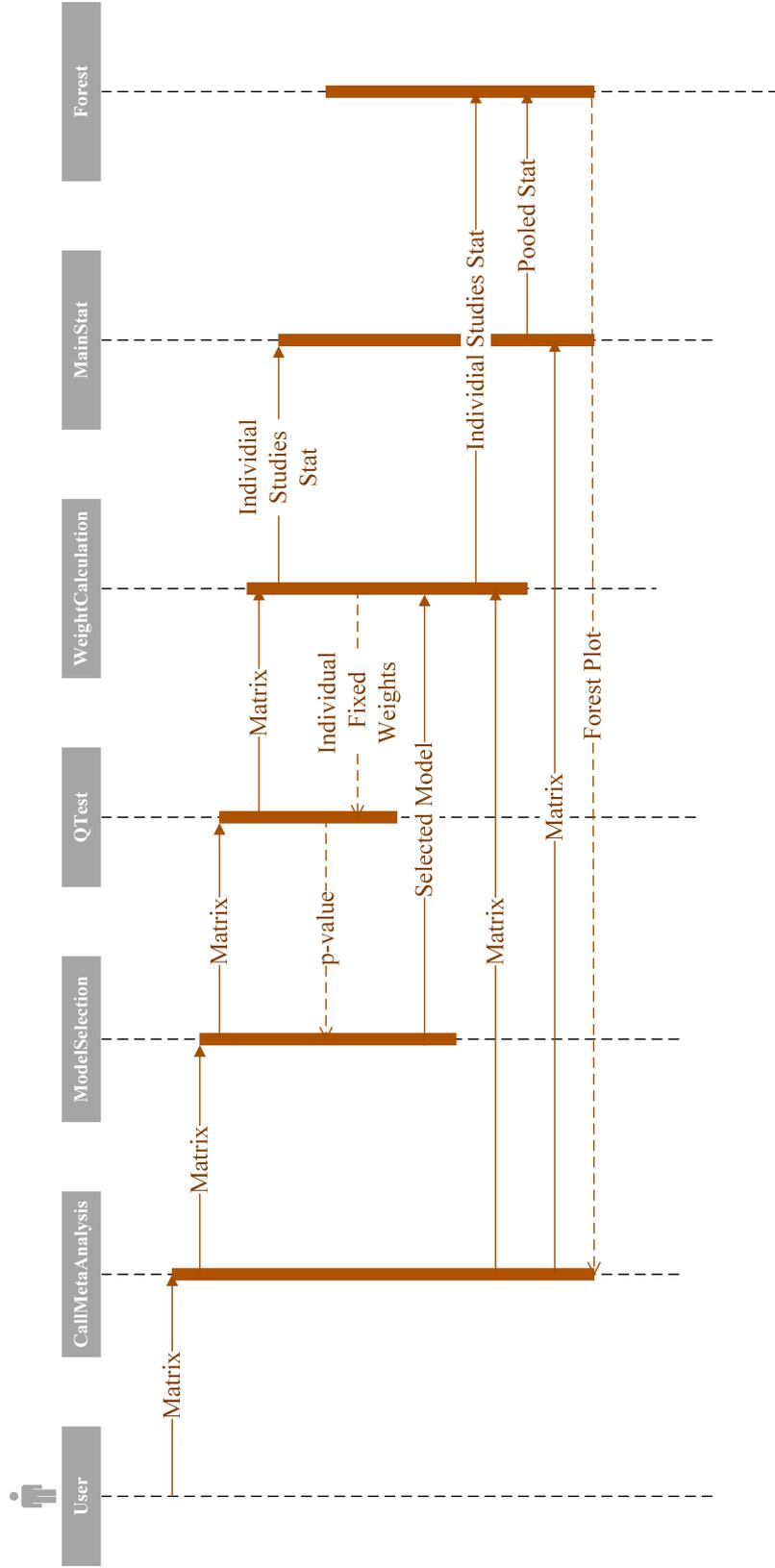


Figure 3.4: UML sequence graph of the Matlab tool for meta-analysis. The objects' blocks represent the functions, the straight arrows the inputs for each function and the dashed arrows the return outputs.

3.2.2.8 Results

According to the search strategy (section 3.2.2.2), 894 titles were identified, 345 in PubMed and 549 in OvidSp. After removing 279 duplicates, 615 titles were considered. Of these, 510 were excluded after reading the abstracts as they did not meet the inclusion criteria. From the remaining 105 abstracts, 77 were removed due to the exclusion criteria. Finally, 28 full-texts were analysed and among these, 16 were excluded due to inclusion/exclusion criteria. Therefore, 12 studies were finally considered appropriate for inclusion in the systematic review. A flowchart of the literature search is shown in Fig. 3.5.

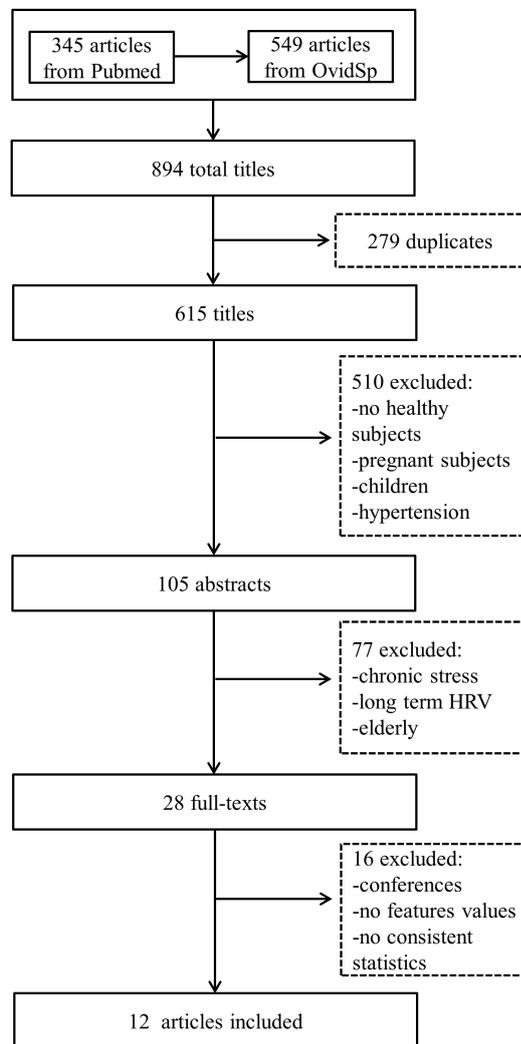


Figure 3.5: Flowchart of literature search: titles, abstract and full-papers included/excluded.

3.2.2.9 Characteristics of the included studies

The 12 studies enrolled from 12 to 399 subjects each, for a cumulative population of 785 subjects. Details on population, HRV analysis reported in each study, HRV length and statistical methods employed to explore significant variations are reported in Table 3.1.

Table 3.1: Characteristics of studies included in the review.

Author, Year	Subjects (Total / women)	HRV analysis	HRV length (min)	Significance tests
Hjortskov <i>et al.</i>, 2004	12/12	Time and frequency	3	ANOVA
Kofman <i>et al.</i>, 2006	30/-	Frequency	10	t-tests, ANOVA
Vuksanovic <i>et al.</i>, 2007	23/13	Frequency and non-linear	5	t-tests
Li. Z <i>et al.</i>, 2009	399/209	Time and frequency	0.5	ANOVA
Schubert <i>et al.</i>, 2009	50/28	Time, frequency and non-linear	3	ANCOVA
Tharion <i>et al.</i>, 2009	18/9	Time and frequency	5	non-parametric statistical tests (Wilcoxon signed rank test)
Papousek <i>et al.</i>, 2010	65/53	Time and frequency	3	ANOVA
Lackner <i>et al.</i>, 2011	20/20	Time and frequency	5	ANOVA
Melillo <i>et al.</i>, 2011	42/-	Non-linear	5	non-parametric statistical tests (Wilcoxon signed rank test)
Taelman <i>et al.</i>, 2011	43/22	Time and frequency	6	non-parametric statistical tests (Wilcoxon and Friedman tests)
Traina <i>et al.</i>, 2011	13/6	Frequency	5	t-test
Visnovcova <i>et al.</i>, 2014	70/39	Time and frequency	6	t-test, non-parametric statistical tests (not specified)

Only three studies used non-parametric statistical tests, which are specifically designed to investigate significant differences in features non-normally distributed [134]. Nine studies [116, 120–123, 125, 126, 128, 129] used ANOVA, ANCOVA or Student's t-test to investigate how HRV features changed before and after the stress section. Of these, Traina *et al.*, Papousek *et al.*, and Lackner *et al.* [125, 126, 128] converted frequency domain HRV features with a log-transform, whereas the remaining studies directly applied standard statistical significance tests to the frequency domain features, which are non-normally distributed, being asymmetric distributions of positive numbers. Another study [122] calculated HF in 0.5 minutes and therefore, it was excluded from the meta-analysis.

The selected studies reported ECG recordings at rest (resting session) and during induced mental stress sessions (stress session). Only one study [124] did not report ECG recordings during the stress session, but the ECG was registered on the day of a university examination, which was assumed to be the stressing event. This may generate heterogeneity, as the response to stress may change during or after a

stress session [135]. Mental stress was induced asking volunteers to perform, under controlled circumstances, one or more of the following tasks: Computer Work Task (CWT), Stroop Colour Word Test (SCWT), Arithmetic task (AT), Video Game Challenge (VGC), Public Speech task (PST), Academic examination (AE), other Physical-Mental Tasks (PMT). A brief description of each mental task is given in Table 3.2. Additionally, in one study [127] the subjects were asked to also perform a physical task. HRV features registered during these physical tasks were not included in this review since they produced a different kind of response to mental stress tasks [136].

Table 3.2: Description of study designs included in the review.

Author, Year	Stressor	Description
Hjortskov <i>et al.</i> , 2004	Computer Work Task (CWT)	The task consisted of keying in random numbers with the dominant hand using the numerical part of the keyboard.
Kofman <i>et al.</i> , 2006	Stroop Colour Word Task (SCWT)	The task required naming the ink colour in which colours' names were printed on a screen being either congruent or incongruent with respect to the colour words.
Vuksanovic <i>et al.</i> , 2007	Arithmetic aloud task (AT)	The task required subtracting 17 from 1000. Subjects were asked to answer as accurate as possible. They told the results after 5 s.
Li. Z <i>et al.</i> , 2009	Video Game Challenge (VGC)	The instructions for video game task 'Breakout' have been standardized via video. The subjects will lie supine on a hospital bed and a monitor 25 inch colour TV was placed 2 metres away.
Schubert <i>et al.</i> , 2009	Speech task (ST)	This task involved preparing and presenting a speech in response to one of two situations. Subjects had 3 min to prepare and 3 min to present their speech.
Tharion <i>et al.</i> , 2009	Academic examination (AE)	The first recording was done on the day of the academic examination and the second during holiday.
Papousek <i>et al.</i> , 2010	Academic examination (AE)	Participants were invited to not move legs and hands while speaking, to read out the questions loud before answering it.
Lackner <i>et al.</i> , 2011	Arithmetical Task (AT)	The task consisted of two triplets of one-digit numbers, which were to be added or subtracted.
Melillo <i>et al.</i> , 2011	Academic Examination (AE)	The first record was performed during verbal examination and the second one was performed after holiday.
Taelman <i>et al.</i> , 2011	Physical-Mental Task (PMT)	The task consisted of continuous mental calculation of five operations with a two or three digit number, which had to be performed without verbal stimulation. Participants used the mouse cursor to indicate the correct answer choosing between three alternatives. The participants would have been rewarded with a movie ticket
Traina <i>et al.</i> , 2011	Arithmetical Task (AT)	The task required to subtracting the number 17 starting from the number 986.
Visnovcova <i>et al.</i> , 2014	Stroop Colour Word Test (SCWT)	The task required to read the colours (green, yellow, orange, red, blue, purple) on the words displayed on a screen, which were congruent or incongruent with the written word.
	Arithmetic Test (AT)	(Results not reported in this review) The task required to calculate three-digit numbers displayed in different random places on the screen into one digit numbers. Subsequently, participants decided that the final result was even or odd by pushing the keyboard arrow.

3.2.2.10 Trends of the HRV features

Tables 3.3, 3.4, 3.5 report the trends and the values of the HRV features in resting and stress sessions in the time, frequency and non-linear domains respectively. Additionally, these tables report the number of subjects enrolled in each study and the relative weight of each study during the pooling.

Time domain features Time domain HRV features reported by the papers shortlisted for this meta-analysis included: the mean of N-N intervals (MeanNN), the standard deviation of an N-N interval (StdNN), the square root of the mean squared difference of successive N-Ns (RMSSD), and a proportion of NN50 divided by the total number of NN values that differ more than 50ms (pNN50) as shown in Table 3.3. In all the studies there was a consensus that the MeanNN, pNN50 and RMSSD decreased during a stress session, although some studies did not demonstrate a significant reduction of these features. A decrease in StdNN was observed in the majority of studies [124, 127, 129] with a high level of significance. Only one study reported contradictory results [123]. In fact, Schubert *et al.* [123] reported a discordant increase in StdNN, which the authors justified as due to a slow respiration rate and a relative reduction in ventilation caused by the speech task used to induce stress in this study.

Table 3.3: Extracted time domain HRV features.

Time domain features	Author, Year	Features trend	Pooling weight of meta-analysis	N	Stress		Rest	
					mean	SD	mean	SD
MeanNN (ms)	Lackner <i>et al.</i> , 2011	↓	1.82%	20	765.31	314.3	837.1	324.5
	Papousek <i>et al.</i> , 2010	↓↓	11.65%	65	617.92	210.0	819.7	244.1
	Schubert <i>et al.</i> , 2009	↓↓	9.25%	50	686.49	240.8	808.6	206.2
	Taelman <i>et al.</i> , 2011	↓↓	19.98%	43	755.44	134.5	863.5	147.1
	Tharion <i>et al.</i> , 2009	↓↓	12.83%	18	777.40	114.3	867.3	114.0
	Visnovcova <i>et al.</i> , 2014	↓↓	41.22%	70	675.99	120.7	847.8	130.4
	Vuksanovic <i>et al.</i> , 2007	↓	3.25%	23	740.74	263.2	806.5	249.5
StdNN (ms)	Schubert <i>et al.</i> , 2009	↑↑	3.50%	50	96.00	86.6	33.20	23.83
	Taelman <i>et al.</i> , 2011	↓↓	37.56%	43	35.40	16.4	46.73	19.48
	Tharion <i>et al.</i> , 2009	↓↓	8.90%	18	52.40	21.7	74.20	25.93
	Visnovcova <i>et al.</i> , 2014	↓↓	50.05%	70	48.98	17.9	56.23	21.67
RMSSD (ms)	Li, Z <i>et al.</i> , 2009	↓↓	26.85%	105	55.50	29.6	68.40	37.70
	Li, Z <i>et al.</i> , 2009	↓↓	14.61%	84	57.20	36.9	74.20	44.90
	Taelman <i>et al.</i> , 2011	↓↓	54.38%	43	19.39	13.8	28.74	16.58
	Tharion <i>et al.</i> , 2009	↓	4.17%	18	49.99	31.07	74.03	39.64
pNN50 (%)	Taelman <i>et al.</i> , 2011	↓	78.43%	43	26.82	16.10	31.83	18.73
	Tharion <i>et al.</i> , 2009	↓↓	21.57%	18	20.57	19.04	39.37	23.79

↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) during stress section ($p > 0.05$).

Frequency domain features Frequency domain HRV features extracted from the shortlisted papers included: low-frequency power (LF), high-frequency power (HF) and LF/HF ratio as shown in Table 3.4. Regarding LF, among the 8 papers reporting this feature, 5 studies agreed that LF increased (3 with statistical significance), while 3 reported an opposite trend. Also, for this case, the findings by Tharion *et al.* [124] were not consistent with the general trend, probably because in this study the stress session was recorded during the day of the examination and

not during the examination session. In addition, Hjortskov *et al.* [120] also showed controversial results, which in this case might be explained by considering that this was the only study in which an introduction session was run before the registration of the resting and stress sessions. Finally, Taelman *et al.* [127] also reported a decreased LF during the stress session, which might also be due to the different protocol adopted, which consisted of physical tasks before the mental stress session. Regarding HF, there is consensus, among the different studies, that this feature decreased during acute mental stress. Only one study indicated a controversial trend, which however was not statistically significant [121]. Regarding the LF/HF ratio, there was a general consensus (5 studies out of 7) that it increased during stress session. The remaining 2 studies [121, 124] reported an opposite trend, which however, is not supported by statistical significance.

Table 3.4: Extracted frequency domain HRV features.

Frequency domain features	Author, Year	Feat. trend	Pooling weight of meta-analysis	N	Stress		Rest	
					mean	SD	mean	SD
LF (ms ²)	Hjortskov <i>et al.</i> , 2004	↓↓	8.52%	12	1391	1028	1664	808
	Lackner <i>et al.</i> , 2011	↑	10.16%	20	1339	1205	812.4	649.9
	Papousek <i>et al.</i> , 2010	↑	14.33%	65	1644	987.0	997.3	598.6
	Taelman <i>et al.</i> , 2011	↓↓	14.83%	43	466.9	460.2	868.4	641.0
	Tharion <i>et al.</i> , 2009	↓↓	5.74%	18	1192	723.6	2155	2157
	Traina <i>et al.</i> , 2011	↑↑	15.36%	13	1241	312.0	511.0	114.5
	Visnovcova <i>et al.</i> , 2014	↑↑	15.69%	70	607.9	457.7	454.9	380.6
	Vuksanovic <i>et al.</i> , 2007	↑↑	15.37%	23	464.0	356.1	387.6	260.3
HF(ms ²)	Hjortskov <i>et al.</i> , 2004	↓↓	5.45%	12	1131	718	1776	1092
	Papousek <i>et al.</i> , 2010	↓	18.14%	65	668.5	401.5	1097	665.1
	Taelman <i>et al.</i> , 2011	↓↓	15.62%	43	552.5	428.0	1005	782.6
	Tharion <i>et al.</i> , 2009	↓↓	1.54%	18	1691	2096	2892	2622
	Traina <i>et al.</i> , 2011	↓	17.91%	13	252.1	263.6	273.1	246.2
	Visnovcova <i>et al.</i> , 2014	↓↓	11.50%	70	445.9	1082	639.1	1337
	Vuksanovic <i>et al.</i> , 2007	↑	9.66%	23	665.1	925.1	595.9	714.4
	LF/HF (-)	Hjortskov <i>et al.</i> , 2004	↑	15.12%	12	1.57	1.09	1.16
Papousek <i>et al.</i> , 2010		↑	22.90%	65	1.11	0.59	0.01	0.80
Schubert <i>et al.</i> , 2009		↑	19.88%	50	1.80	1.20	1.50	1.11
Tharion <i>et al.</i> , 2009		↓	12.74%	18	1.30	0.80	1.40	1.80
Traina <i>et al.</i> , 2011		↑↑	4.61%	13	5.81	2.88	2.57	2.10
Vuksanovic <i>et al.</i> , 2007		↓	4.61%	23	1.07	3.83	1.08	2.92
Kofman <i>et al.</i> , 2006		↑	20.14%	30	1.48	1.02	0.97	0.68

↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) during stress section ($p > 0.05$).

Non-linear features Non-linear HRV features extracted from the shortlisted papers included: the Shannon and the Sample Entropy (respectively ShanEn and SampEn), the largest Lyapunov exponent (LLE), the correlation dimension (D2), short- and long-term fluctuation slope (dfa1 and dfa2), the standard deviation of the Poincaré plot perpendicular and along to the line of identity (respectively SD1 and SD2), Recurrence Plot determinism (RPadet) and Recurrence Rate (REC),

maximal length of lines (RPI_{\max}) and mean length of lines (RPI_{mean}). Finally, the Approximate Entropy (ApEn) computed with the threshold r set to $0.2 \cdot \text{StdNN}$, to the value maximising the entropy and to the value computed by Chon [137], and reported respectively as ApEn (0.2), ApEn(rmax) and ApEn(chon) as shown in Table 3.5. Only two features were investigated by more than one study: D2 [45, 123] was reported as consistently reduced during a stress session (an academic examination in Melillo *et al.* [45] and an arithmetical task in Schubert *et al.* [123]); dfa1 [45, 121] which was reported with significantly opposite trends.

Table 3.5: Extracted non-linear HRV features.

Non-linear domain features	Author, Year	Features trend	Pooling weight of meta-analysis	N	Stress		Rest	
					Mean	SD	Mean	SD
dfa1 (-)	Melillo <i>et al.</i> , 2011	↓↓	49.41%	42	1.05	0.44	1.41	0.16
	Vuksanovic <i>et al.</i> , 2007	↑↑	50.59%	23	0.97	0.19	0.85	0.19
D2 (-)	Melillo <i>et al.</i> , 2011	↓↓	45.89%	42	1.65	1.28	2.83	1.09
	Schubert <i>et al.</i> , 2009	↓↓	54.11%	50	3.2	0.33	3.5	0.27
SD1 (ms)	Melillo <i>et al.</i> , 2011	↓↓	-	42	0.02	0.01	0.02	0.01
SD2 (ms)	Melillo <i>et al.</i> , 2011	↓↓	-	42	0.05	0.02	0.08	0.02
ApEn (0.2) (-)	Melillo <i>et al.</i> , 2011	↓↓	-	42	0.99	0.24	1.09	0.13
ApEn (r_{chon}) (-)	Melillo <i>et al.</i> , 2011	↓↓	-	42	0.98	0.24	1.11	0.11
ApEn (r_{\max}) (-)	Melillo <i>et al.</i> , 2011	↓	-	42	1.09	0.17	1.12	0.10
dfa2 (-)	Melillo <i>et al.</i> , 2011	↓	-	42	0.76	0.13	0.78	0.18
ShanEn (-)	Melillo <i>et al.</i> , 2011	↑↑	-	42	3.42	0.39	3.17	0.23
RPadet (%)	Melillo <i>et al.</i> , 2011	↑	-	42	98.75	1.28	98.61	0.86
REC (%)	Melillo <i>et al.</i> , 2011	↑↑	-	42	42.24	12.05	33.46	6.27
RPI_{mean} (Beats)	Melillo <i>et al.</i> , 2011	↑↑	-	42	14.88	6.77	11.09	2.48
RPI_{\max} (Beats)	Melillo <i>et al.</i> , 2011	↓↓	-	42	213.4	136.5	286.7	111.2
LLE	Vuksanovic <i>et al.</i> , 2007	↑	-	23	0.06	0.019	0.06	0.019
SampEn (-)	Vuksanovic <i>et al.</i> , 2007	↓↓	-	23	1.65	0.06	1.77	0.04

↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) during stress section ($p > 0.05$).

3.2.2.11 Pooled pivot values of HRV features

From the included studies, 8 HRV features were pooled in this systematic meta-analysis because they were reported at least in two papers: 4 features in the time domain (MeanNN, StdNN, pNN50, RMSSD), 3 in the frequency domain (LF, HF, LF/HF) and 2 in the non-linear domain (dfa1, D2). The relative pooling weights for each study are reported in the Tables 3.3, 3.4 and 3.5, for the time, frequency and non-linear domain respectively. The results of the pooling are reported in Table 3.6, where the trends of the pooled HRV features are also reported.

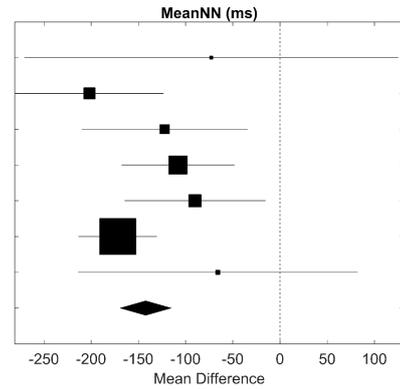
Table 3.6: Pooled HRV features.

Outcome	Subjects	Heterogeneity I^2, p	Model	Units	MD	CI95%	p-value	Trend
<i>Time</i>								
MeanNN	289	33%, 0.17	Fixed	ms	-142.2	(-168.9; -115.47)	<0.01	↓↓
StdNN	181	92%, <0.01	Fixed	ms	-7.627	(-12.20; -2.97)	<0.01	↓↓
RMSSD	250	0%, 0.50	Fixed	ms	-12.03	(-16.78; -7.28)	<0.01	↓↓
pNN50	61	65%, 0.09	Fixed	-	-7.98	(-14.52; -1.45)	<0.05	↓↓
<i>Frequency</i>								
LF	264	91%, <0.01	Random	ms ²	156.1	(-157.6; 469.8)	0.33	↑
HF	244	62%, <0.01	Random	ms ²	-256.6	(-376.8; -154.43)	<0.01	↓↓
LF/HF	211	75%, <0.01	Random	-	0.61	(0.14; 1.08)	<0.01	↑↑
<i>Non linear</i>								
dfa1	65	96%, <0.01	Random	-	-0.12	(-0.59; 0.35)	0.63	↓
D2	92	91%, <0.01	Fixed	-	-0.35	(-0.46; -0.23)	<0.01	↓↓

↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) during stress section ($p > 0.05$); MD: Mean Difference CI95%: Confidence Interval at 95%; '-': dimensionless pooled HRV features.

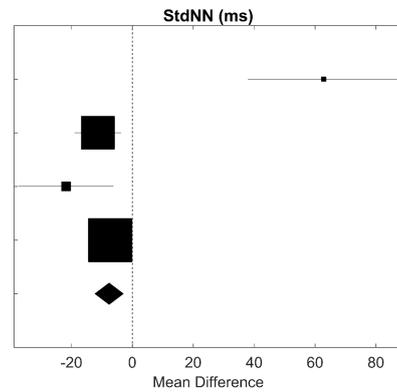
In Figs. 3.6, 3.6a and 3.6b the Forest plots are reported.

Study	MD	Low	High	p-val
Lackner, 2011	-72.683	-270.683	125.317	0.476
Papousek, 2010	-201.752	-280.036	-123.468	0.000
Schubert, 2009	-122.126	-209.970	-34.282	0.008
Taelman 2011	-108.020	-167.785	-48.255	0.001
Tharion, 2009	-89.900	-164.478	-15.322	0.024
Visnovcova, 2014	-171.790	-213.399	-130.181	0.000
Vuksanovic, 2007	-65.711	-213.918	82.496	0.390
Pooled	-142.185	-168.900	-115.470	0.000



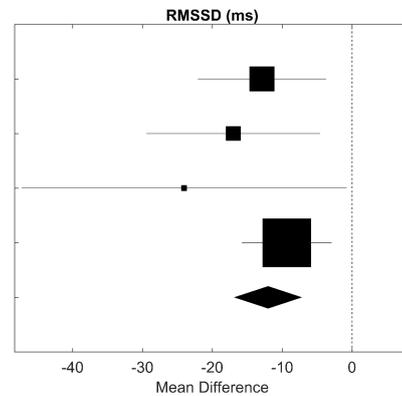
(a) MeanNN Forest plot.

Study	MD	Low	High	p-val
Schubert, 2009	62.800	37.893	87.707	0.000
Taelman, 2011	-11.330	-18.932	-3.728	0.004
Tharion, 2009	-21.800	-37.411	-6.189	0.010
Visnovcova, 2014	-7.250	-13.834	-0.666	0.033
Pooled	-7.627	-12.286	-2.969	0.001



(b) StdNN Forest plot.

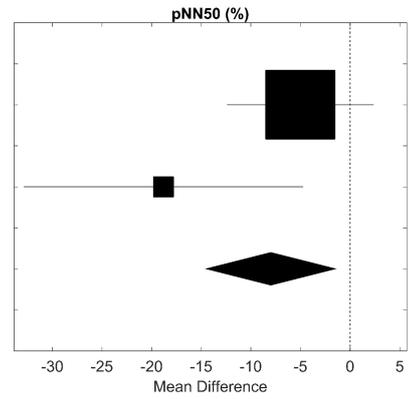
Study	MD	Low	High	p-val
Li, Z, 2009	-12.900	-22.068	-3.732	0.006
Li, Z, 2009	-17.000	-29.429	-4.571	0.008
Taelman, 2011	-24.040	-47.309	-0.771	0.051
Tharion, 2009	-9.350	-15.792	-2.908	0.006
Pooled	-12.033	-16.783	-7.282	0.000



(c) RMSSD Forest plot.

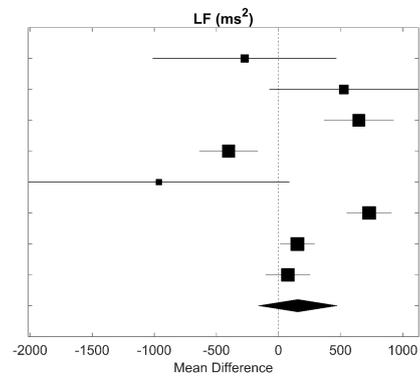
Figure 3.6: Forest plots of the pooled HRV features. MD: Mean Difference; Low and High represent the range of the confidence interval; p-val: p-value between rest and stress sessions.

Study	MD	Low	High	p-val
Taelman, 2011	-5.010	-12.392	2.372	0.187
Tharion, 2009	-18.800	-32.877	-4.723	0.013
Pooled	-7.985	-14.522	-1.447	0.021



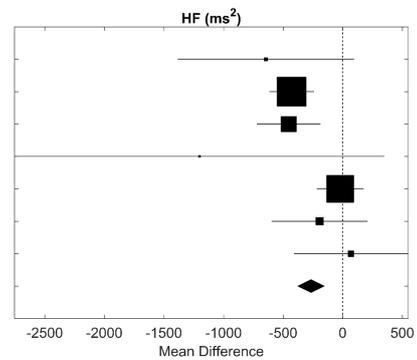
(d) pNN50 Forest plot.

Study	MD	Low	High	p-val
Hjortskov, 2004	-273.000	-1012.808	466.808	0.477
Lackner, 2011	527.025	-73.196	1127.247	0.093
Papousek, 2010	646.936	366.311	927.562	0.000
Taelman, 2011	-401.550	-637.417	-165.683	0.001
Tharion, 2009	-962.800	-2014.035	88.435	0.082
Traina, 2011	730.117	549.469	910.765	0.000
Visnovcova, 2014	153.029	13.576	292.482	0.033
Vuksanovic, 2007	76.444	-103.808	256.695	0.410
Pooled	156.082	-157.590	469.754	0.33



(e) LF Forest plot.

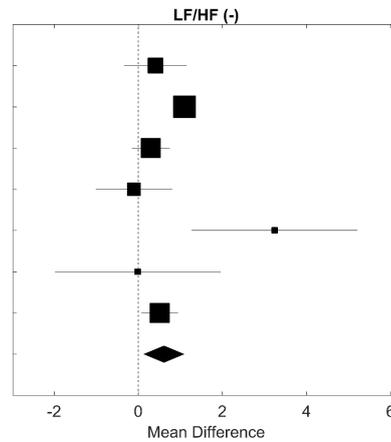
Study	MD	Low	High	p-val
Hjortskov, 2004	-645.000	-1384.448	94.448	0.101
Papousek, 2010	-428.157	-617.040	-239.275	0.000
Taelman, 2011	-453.020	-719.629	-186.411	0.001
Tharion, 2009	-1201.250	-2752.255	349.755	0.138
Traina, 2011	-21.000	-217.096	175.095	0.836
Visnovcova, 2014	-193.203	-596.045	209.638	0.349
Vuksanovic, 2007	69.285	-408.398	546.968	0.778
Pooled	-265.616	-376.801	-154.431	0.000



(f) HF Forest plot.

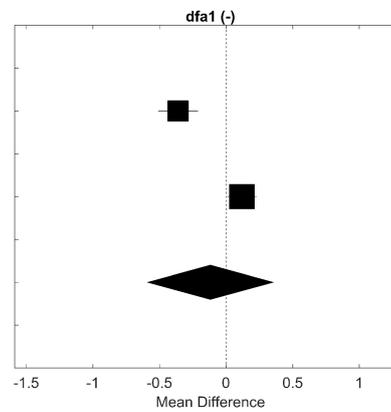
Figure 3.6a: Forest plots of the pooled HRV features (cont.). MD: Mean Difference; Low and High represent the range of the confidence interval; p-val: p-value between rest and stress sessions.

Study	MD	Low	High	p-val
Hjortskov, 2004	0.410	-0.335	1.155	0.293
Papousek, 2010	1.100	0.858	1.342	0.000
Schubert, 2009	0.300	-0.153	0.753	0.197
Tharion, 2009	-0.100	-1.010	0.810	0.831
Traina, 2011	3.240	1.268	5.212	0.004
Vuksanovic, 2007	-0.010	-1.982	1.962	0.992
Kofman, 2006	0.512	0.075	0.949	0.025
Pooled	0.613	0.143	1.083	0.009



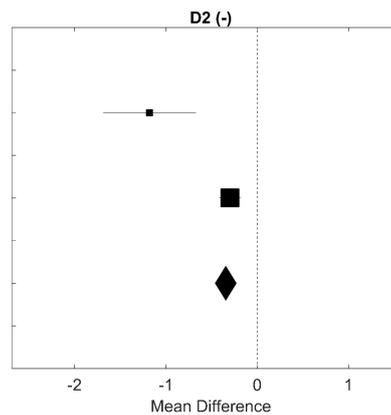
(g) LF/HF Forest plot.

Study	MD	Low	High	p-val
Melillo, 2011	-0.360	-0.510	-0.210	0.000
Vuksanovic, 2007	0.120	0.010	0.230	0.038
Pooled	-0.117	-0.588	0.353	0.63



(h) dfa1 Forest plot.

Study	MD	Low	High	p-val
Melillo, 2011	-1.180	-1.688	-0.672	0.000
Schubert, 2009	-0.300	-0.418	-0.182	0.000
Pooled	-0.345	-0.460	-0.230	0.000



(i) D2 Forest plot.

Figure 3.6b: Forest plots of the pooled HRV features (cont.). MD: Mean Difference; Low and High represent the range of the confidence interval; p-val: p-value between rest and stress sessions.

3.2.2.12 Models to detect mental stress via short HRV features

Although in the existing literature there are many studies that developed models to detect mental stress using short term HRV features, only two studies ([45] and [128]) included in this review proposed a model to automatically detect mental stress.

Melillo *et al.* [45] proposed a model based on Linear Discriminant Analysis (LDA), employing three HRV nonlinear features: SD1, SD2 and ApEn(0.2). The proposed model achieved accuracy, sensitivity and specificity respectively of 90%, 86% and 95% in automatically detecting subjects under stress. These performances were achieved testing the developed classifier using a 10-fold-cross validation technique, based on the subjects' exclusions. Traina *et al.* [128] studied the Pearson's correlation between the frequency domain features (high and low components of the power spectrum) before and after the stress session, demonstrating that those correlations were significant. However, this is partially arguable as the Pearson's correlation [138] would eventually lie on the assumption that the HRV features are normally distributed, whereas HRV frequency features are not [15]. This paper did not develop a predictive model and did not validate the correlation with a cross-validation technique.

3.2.2.13 Discussion

This first part of this chapter presented the results of a systematic literature review with meta-analysis of the articles investigating how short HRV features changed during induced acute mental stress. In this review, 12 studies were shortlisted and changes in 22 HRV features during mental stress were systematically reported. Finally, 9 HRV features were pooled.

The results demonstrated that 4 HRV features (MeanNN, RMSSD, pNN50 and D2) decreased during stress [122–124, 126, 127, 129]. The majority of studies [120, 124, 126–129] agreed that StdNN and HF decreased during stress, while LF/HF and LF increased during stress [116, 120, 121, 123, 125, 126, 128, 129]. Regarding StdNN, only one study [123] out of the 4 [123, 124, 127, 129] considering this feature, reported an increasing value during stress. This may be due to the fact that in [123] the authors analysed HRV excerpts of 3 minutes, without assessing the validity of ultra-short HRV features, while in [124, 127, 129] the authors analysed the standard 5 minutes HRV excerpts. Regarding LF, 3 studies [120, 124, 127] out of 8 [120, 121, 124–129] considering this feature, reported a decreasing value during stress. However, these studies adopted study designs significantly different from the others. Different from the others, the first two studies used physical movements during

the stress session: in Hjortskov *et al.* [120], participants used the dominant hand to digit random numbers on the left part of a keyboard; in Taelman *et al.* [127], participants used the mouse cursor to indicate the correct answer choosing between three alternatives. Different from the other protocols, the use of the hand activated a cortical area that is not triggered by the designs adopted by the remaining studies, where there is no physical activity. These results were also consistent with the findings reported in Yu *et al.* [139] with arithmetic test, in which the participants were required to use the keyboard. The other study reporting a decreased value of LF during the stress session was Tharion *et al.* [124], which, however, measured the physiological response to the mental stress during the day of the University examination, and not during the examination, as done in [45, 126], which also used academic examination as stressor. In this regard, there is a consensus that the reaction to a stressful situation is composed of several phases [135], each implying a different response of the ANS and therefore, a different HRV modulation. Finally, only one non-linear HRV feature, dfa1, reported by two studies [45, 121], achieved completely opposite results, and both with statistical significance. No clear explanation for such heterogeneous results can be inferred by the two papers.

Regarding the meta-analysis, the pooled values of 7 HRV features (MeanNN, StdNN, RMSSD, pNN50, D2, HF and LF/HF) out of the 9 meta-analysed changed significantly during mental stress. The pooled values of those 7 features should be regarded as they are more reliable than those presented by each paper and therefore, considered as pivot values for the studies presented in Chapter 5. For instance, the increase of the pooled LF/HF was statistically significant whereas it was not statistically significant in 6 out of the 7 papers reporting this HRV features. Since the p-value is strongly dependent on the sample size, a possible explanation is that the number of volunteers enrolled in each of those 6 studies was too small compared with the LF/HF mean differences measured, which, in turn, were too small compared with the standard deviations measured in each group during stress and rest [131]. Regarding the LF, the pooled results would have become statistically significant if the three studies employing a different design (as discussed above, [127] and [120] moving hands, [124] measuring stress in the day of examination and not during it) were excluded. In fact, removing these 3 studies the pooled mean difference for LF during stress was 286.56 ms^2 (CI 95% 183.89-389.23, p-value<0.01, fixed effect model), calculated on 164 subjects with very low heterogeneity. Therefore, more studies should investigate the effect of moving hands or low physical activity during stress sessions.

Finally, it is hard to discuss the results for dfa1, since the two studies calculating

this HRV feature adopted similar protocols and comparable sample sizes. Melillo *et al.* [45] enrolled approximately double the number of subjects, but the SD measured in the stress session was too high, affecting the weight of the study. Moreover, only a few studies [45, 121, 123] investigated the non-linear HRV features and their behaviours during mental stress. Furthermore, due to the low number of subjects enrolled in these studies [45, 121, 123] the outcomes are not easily comparable with those for the linear HRV analysis. Therefore, further studies investigating non-linear HRV features during mental stress are still needed.

The decreases of MeanNN, RMSSD, pNN50, StdNN reflected a depressed HRV during stress. This is consistently confirmed by the pooled values of HRV frequency features, among which HF proved to decrease significantly, reflecting a decreased HRV variability. Moreover, the observed decrease in D2 during stress sessions can be interpreted as a reduced complexity of the HRV, reflecting lower adaptability and fitness of the cardiac pacemaker and a functional restriction of the participating cardiovascular elements [123]. Finally, the frequency domain features consistently supported the idea that during stress there is a general depression in HRV with a relative displacement of the vago-sympathetic balance, during which the sympathetic activation relatively overcomes the parasympathetic one. In fact, the LF accounting for activation of both the parasympathetic and sympathetic system, increased, while the HF, that is associated with the parasympathetic system activation, decreased. This result was confirmed by the increase in LF/HF. These outcomes confirmed the induced shift of the ANS balance towards the sympathetic activation and the parasympathetic withdrawal during acute mental stress [129]. This phenomenon was explained through the theory of the fight or flight response, which supports the idea that there is an inhibition of the vagus and a prevalence of the sympathetic system during a stressful situation [135]. This result could change if the stressing session was measured after (and not during) the stress event, as demonstrated in [124] and consistently with the phases of the stress as described in [135].

3.2.2.14 Conclusion and recommendations

In conclusion, this review proved that there was a consensus on the HRV features changing consistently during mental stress. Particularly, the results of the pooled HRV features provided pivot values for at least 7 HRV features that changed significantly during stressing sessions. These significant changes confirmed previous results about the induced shift of the ANS balance towards the sympathetic activation and the parasympathetic withdrawal during acute mental stress. However, huge heterogeneity among studies investigating the physiological response to men-

tal stress was observed in this review. Moreover, this review identified gaps in the existing literature proving that future studies are needed to confirm the behaviour of non-linear HRV features and ultra-short HRV features during stress. In fact, only 3 out of the 12 studies investigated non-linear HRV analysis and only 4 out of the 12 studies investigated HRV recordings of less than 5 min .

Finally, this review informed the study designs for the studies presented in the following chapters. In fact, future studies are recommended to: clearly define the study design (i.e., length of HRV features) and the study protocol (i.e., definition of stressor, avoid high physical activity if not required) according to the best available evidence; to analyse HRV features accordingly to international guidelines; to use statistical tests consistent with the HRV measure distribution (i.e., check if HRV features are normally distributed or not); to check HRV stationarity if HRV features are longer than 5 minutes; to measure stress session according to the goal of the study (i.e., during, before or after the stress session).

Moreover, based on the results of this review, it was clear that not many studies have investigated ultra-short HRV features to detect stress and some studies extracted HRV features in excerpts shorter than 5 minutes without assessing the validity of the latter. This was mainly due to the lack of clear guidelines on how to analyse HRV in the ultra-short term. Therefore, a review of the existing methods used to assess the validity of ultra-short HRV features as good surrogates of short term ones, not only to detect stress, was carried out and presented in the next section (deliverable 1b).

3.2.3 Literature review on methods to assess ultra-short HRV features

As demonstrated in the previous sections, only a few studies have investigated mental stress using ultra-short HRV features and fewer studies questioned the validity of ultra-short HRV features. Therefore, a literature review of the existing methods used to assess the validity of ultra-short HRV features was carried out.

In e-health monitoring the conventional 5 minute recordings might be unsuitable, due to real-time requirements. In fact, ultra-short term HRV analysis, especially in combination with wearable sensors, may allow continuous and real-time monitoring of an individual's stress level, which may be important in some circumstances or professions (e.g., surgeons, aeroplane pilots), but numerous challenges have arisen from shortening HRV excerpts below 5 minutes.

Many apps and wearable devices aim to perform stress analysis in real time [140]. For instance, the StressAware app [16], the SmartCoping app [141], the

ithlete app [20], the NeuroSky technology [19], the PulseOn wristband [18], the Tink device [17], and many others are being released onto the market, claiming to do HRV analysis in real time (from 10 sec to 1 min). Although there is a clear need for such technologies, unfortunately, two problems remain unsolved: there are not yet clear guidelines on how to analyse HRV in the ultra-short term; there is not a clear framework to identify reliable subsets of ultra-short HRV features for the automatic detection of stress.

Long term (nominally 24 hours) and short term (nominally 5 min) HRV features have been widely investigated, physiologically justified and clear guidelines for analysing HRV in 5 min or 24 hours recordings are available, whereas for ultra-short term HRV there are no valid methods to assess its reliability. From the theoretical point of view, it should be well-known that some HRV features are not computable in ultra-short term [15]. For instance, it is generally recommended that spectral analyses are performed on recordings lasting for at least 10 times more than the slowest significant signal oscillation period. In the case of short term HRV analysis, the slower significant oscillations in the so-called Low Frequency (LF) power spectrum bandwidth have a frequency of 0.04 Hz and therefore, a period of 25 sec. Thus, in order to measure the LF power spectrum of HRV excerpts, at least 250 sec length HRV signals are required. In the same manner, in order to compute the High Frequency (HF) power, at least 1 min is required. Therefore, it is not possible to compute LF and HF power spectrum with HRV excerpts shorter than 1 min. As far as non-linear HRV features are concerned, less has been explored in the existing literature. However, the approximate entropy (ApEn) measure has shown to be unreliable in excerpts lasting less than 3 min [142].

3.2.3.1 Methods and materials

Relevant studies on the use of ultra-short HRV analysis were first identified and selected by searching on the PubMed and OvidSP databases. Articles were searched using Boolean combinations of the following keywords or their equivalent Medical Subject Heading (MeSH) terms: “Heart Rate Variability”, “HRV”, “ultra-short, “very short”. Title, abstract and full text were chosen as fields for the search. However, due to the lack of guidelines on how to analyse HRV in ultra-short term, the nomenclature used in many scientific papers is very heterogeneous, if not misleading. For instance, many studies performing HRV analysis on segments shorter than 5 min, did not use the tag “ultra-short term” or did not mention the length of HRV excerpts analysed (i.e., ultra-short, short or long term analysis). Therefore, a linear search of references of the retrieved articles was required and performed. The heterogeneity

and quality of the available literature led me to conduct a state-of-art review to address the current concerns.

To limit the linear search, the following criteria were utilised: papers published in the last 15 years (since 2003), focusing on healthy and non-pregnant adult humans. Shortlisted papers were considered suitable for this review if they met the following criteria:

- the subjects were human beings over 18 years old;
- HRV was analysed on excerpts shorter than 5 minutes;
- HRV features were extracted with standardised methods as described in [15].

Since guidelines on ultra-short HRV analysis were still missing and a clear physiological interpretation of ultra-short HRV features was not available, the retrieved literature was analysed in agreement with medical literature on surrogate outcomes [143, 144], which is based on three main requirements: valid surrogate must be correlated with the clinical endpoint (benchmark); a valid surrogate should capture a reliable and sufficiently large portion of the treatment effect on the clinical endpoint; a valid surrogate should be able to predict the treatment effect on the clinical endpoint. More details are discussed in Chapter 4, section 4.2.1.

Short term HRV analysis was assumed as the benchmark in this review.

3.2.3.2 Results and discussion

Since 2003, 29 papers [120, 122, 123, 126, 140, 145–168] have been identified. Those studies focused on ultra-short HRV features for different purposes: 18 focused on mental stress or mental workload detection [120, 122, 123, 126, 140, 145, 147–150, 153, 159–161, 164, 166–168]; one focused on athletic performance monitoring [151]; one focused on auditory stimuli [157]; the remaining investigated the reliability of ultra-short HRV features in a control condition (e.g., only a resting condition). The 18 studies investigating mental stress used one or more of the following tasks to induce mental stress: a Computer Work Task (CWT), a Flight Simulator (FS), a Stroop Colour Word Task (SCWT), an Arithmetic Task (AT), a Memory Task (MT), a Logic Task (LT), a Video Game Challenge (VGC), a Public Speech Task (PST), an Academic Examination (AE), some other Physical-Mental Task (PMT). Some papers [120, 122, 149–152, 156, 157, 165, 168] investigated only three or fewer HRV features. Choi *et al.* [149] investigated frequency HRV features to detect stress using 240 sec excerpts. De Rivecourt *et al.* [150] explored MeanHR, LF and HF as indices for momentary changes in mental effort during simulated flight. However,

only HRV frequency features were investigated at different lengths (i.e., 240, 120, 60 and 30 sec). Esco *et al.* [151] only investigated RMSSD during pre and post exercise, as RMSSD proved to be a reliable feature to assess performance in athletes. They investigated RMSSD at different time scales (10, 30, and 60 sec). Flatt *et al.* [152] also investigated one HRV feature, RMSSD at 55 sec during a control condition. Hjortskov *et al.* [120] explored only HRV frequency features (LF, HF and LF/HF) in 3 min segments during rest and working at a computer. Li *et al.* [122] investigated MeanNN, RMSSD and HF over 30 sec compared to the total duration of rest and stress sessions (i.e., 10 min). Munoz *et al.* [156] also investigated only two HRV features: StdNN, RMSSD at 10, 30 and 120 sec during a control condition. Nardelli *et al.* [157] investigated Poincaré plot features (SD1, SD2) at 15, 25 and 60 sec during a control condition and effective sound. Thong *et al.* [165] investigated three HRV features StdNN, RMSSD, HF at different time scales from 10 to 300 sec (with a step of 10 sec) during a control condition. Wang *et al.* [168] investigated MeanNN, RMSSD and HF during rest and stress sessions at 30 sec. The remaining studies investigated more than three HRV features.

An overview of the methods employed in the 29 shortlisted papers to assess the validity of ultra-short HRV features is synthetically reported in Fig. 3.7, whereas the characteristics of the reviewed studies are reported in Tables 3.7 and 3.7a.

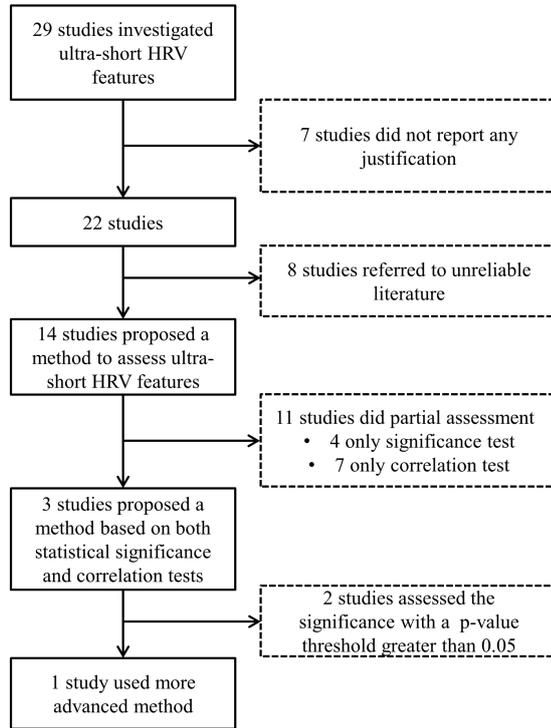


Figure 3.7: Flowchart of literature review. An overview of methods employed in the 29 shortlisted papers to assess the validity of ultra-short HRV features.

As shown in Fig. 3.7, 7 of the 29 studies [120, 123, 126, 145, 149, 159, 166] did not report any method used to validate the use of ultra-short HRV features or reference to support the adoption of ultra-short HRV features. Eight other studies [140, 147, 152–154, 164, 167, 168] also did not report any method used to validate the use of ultra-short HRV features but they relied on the results of five previous studies [150, 158, 161–163], which cannot be considered fully reliable as detailed below. Eleven studies [122, 148, 150, 155, 157, 158, 160–163, 165], including the five mentioned in the previous sentence [150, 158, 161–163], performed only a partial assessment either only using statistical significance or only performing correlation tests. In fact, three of those eleven studies [161, 162, 165] employed statistical significance tests to prove that there were no statistically significant changes in HRV features during the short VS ultra-short term, assuming short term HRV analysis (i.e., 5min) as the benchmark. They concluded that ultra-short HRV features were good surrogates of short term ones, if no-significant differences were observed, using a significance threshold greater than 0.05 ($p\text{-value} > 0.05$).

Unfortunately, this result is arguable because, while a $p\text{-value} < 0.05$ is con-

ventionally used to support the hypothesis that two distributions are significantly different, it is well-known that no conclusions can be drawn for p-value greater than 0.05, as detailed in [169]. For instance, two distributions could result in a p-value greater than 0.05 because of their cardinalities. However, one of those three studies [161] also assessed in a second analysis ultra-short term HRV features for two conditions (i.e., rest and stress) using a non-parametric test (p-value<0.05) to find the shortest duration that distinguished between the two conditions. Nevertheless, in this case the results are also arguable as this study [161] relied on the results conducted on the first analysis to prove that there were no statistically significant changes in HRV features in short VS ultra-short term using a p-value greater than 0.05. Furthermore, one study [160] used one-way ANOVA to determine which ultra-short HRV feature (i.e., 220, 150, 100 and 50 sec) could discriminate between rest and stress sessions (p-value<0.05) . However, due to the nature of HRV features, which are non-normally distributed (especially in the frequency domain), a non-parametric test should have been used instead.

Analogically, 7 studies [122, 148, 150, 155, 157, 158, 163] employed only correlation tests to prove that ultra-short term HRV features behaved as short term; in fact, they concluded that ultra-short HRV features were good surrogates if significantly correlated with their equivalent short HRV features. This result is arguable because, as stated by Fleming *et al.* [143], “a correlate does not make a surrogate”, although it is the first step for the identification of a good surrogate.

Only two studies [146, 151] performed both statistical significance tests and correlation analysis. Unfortunately, also in these two studies, the statistical significance analysis consisted only of observing if the p-value was greater than 0.05, which is not a suitable method as discussed above.

Some studies employing invalid statistical significance analysis reported misleading results, especially regarding HRV frequency features. Baek *et al.* [146] and Salahuddin *et al.* [162] computed VLF in 270 sec and 50 sec excerpts although, as reported also in [15], VLF is only reliable in long term HRV analysis. De Rivecourt *et al.* [150], who employed only correlation analysis, and Salahuddin *et al.* [162], who employed an arguably statistical significance test, reported that LF and HF are reliable in segments lower than 30 sec, while, as already discussed and also stated in [15], at least 250 and 60 sec are necessary for LF and HF respectively.

Finally, only one study [156] investigated the validity of ultra-short HRV features in a more rigorous way. In fact, Munoz *et al.* [156] compared 10, 30, and 120 sec HRV features with 5 min ones, using Pearson's correlation test (after having normalised HRV features with log-transformation), Bland-Altman plots [170] and

Cohen's d statistical test [171]. Unfortunately, in this study, the authors reported the results of only 2 time-domain HRV features under one standard condition (i.e., resting).

Hence, among the 29 identified papers, 1 paper justified the adoption of ultra-short HRV features with a rigorous method but focused only on 2 time-domain HRV features. Conversely, 7 papers did not provide any justification, 8 papers based their choice on unreliable articles, 11 papers performed only a partial assessment (i.e., either statistical significance or correlation tests) and 2 papers performed a complete assessment (both statistical significance and correlation tests) but using statistical significance tests improperly. Moreover, eight studies [120, 147, 149, 154, 158–161] enrolled a very low number of subjects, therefore, no firm conclusions can be drawn from their results. Overall, none of those 29 studies proposed a valid method to identify reliable subsets of ultra-short HRV features or surrogates of the short term HRV features to allow the detection of the event of interest (e.g., stress). Therefore, future studies in this area are required.

Consequently, this short review demonstrated that there is a clear lack of rigorous methods to assess the validity of ultra-short HRV features and to identify reliable subsets of ultra-short HRV features to detect mental stress using ultra-short term HRV analysis to enable reliable real-time stress detection using wearable sensors and portable devices. In other words, the reviewed literature highlighted that some valuable methodologies are available and already in use, but in a very fragmented way, resulting in inappropriate or inaccurate practices.

Table 3.7: Characteristics of studies investigating ultra-short HRV.

Author, Year	Physiological signals	HRV features investigated	Length (sec)	Condition	N. Sub	Justification for ultra-short HRV adoption
Arza <i>et al.</i> , 2015	HRV	MeanHR, StdNN, RMSSD, pNN50, VLF, LF, HF, LF/HF and LFnu	180	Rest/Stress	25	None
Baek <i>et al.</i> , 2015	HRV	MeanHR, StdNN, RMSSD, pNN50, VLF, LF, HF, LF/HF, TotPow, , LFnu and HFnu	270 to 10	Control	500	<ul style="list-style-type: none"> • Stat.: Kruskal–Wallis test (p-val>0.05) • Cor.: Pearson's correlation analysis and Bland–Altman plot
Boonnithi <i>et al.</i> , 2011	HRV	MeanNN, StdNN, MeanHR, StdHR, RMSSD, pNN50, VLF, LF, HF, LF/HF, LFnu, HFnu	50	Rest/Stress	6	<ul style="list-style-type: none"> • Referred to literature
Brisinda <i>et al.</i> , 2014	HRV, BP	All features reported in Table I, except HRV index and TINN	120, 60, 30	Rest/Stress	113	<ul style="list-style-type: none"> • Cor.: ICC
Choi <i>et al.</i> , 2009	HRV	LF, HF, LF/HF	240	Rest/Stress	3	<ul style="list-style-type: none"> • None
De Rivecourt <i>et al.</i> , 2008	HRV, eye activity	MeanHR, LF and HF	240, 120, 60, 30	Rest/mental workload	19	<ul style="list-style-type: none"> • Cor.: Pearson's on log transformed features
Esco <i>et al.</i> , 2014	HRV	RMSSD	60, 30, 10	Pre/Post exercise	23	<ul style="list-style-type: none"> • Stat.: ANOVA (p>0.05), Cohen's d • Cor.: ICC and Bland–Altman graph on log transformed features
Flatt <i>et al.</i> , 2013	HRV	RMSSD	55	Control	25	<ul style="list-style-type: none"> • None-referred to literature
Hjortskov <i>et al.</i> , 2004	HRV, BP	LF, HF and LF/HF	180	Rest/Stress	12	<ul style="list-style-type: none"> • None
Kim <i>et al.</i> , 2008	HRV	StdNN, RMSSD, pNN50, HRV index, TINN, LF, HF	180	Rest/Stress	68	<ul style="list-style-type: none"> • None-referred to literature
Kwon <i>et al.</i> , 2016	HRV	StdNN, RMSSD, MeanHR, LF, HF, LF/HF, TotPow, LFnu and HFnu	30	Control	14	<ul style="list-style-type: none"> • None-referred to literature
Li <i>et al.</i> , 2009	HRV	MeanNN, RMSSD and HF	30	Rest/Stress	399	<ul style="list-style-type: none"> • Cor.: Pearson on log transformed features
Mayya <i>et al.</i> , 2015	HRV	StdNN, RMSSD, pNN50, LF, HF, LF/HF, SD1, SD2, and dfal	60	Rest/Stress	49	<ul style="list-style-type: none"> • None-referred to literature
McNames <i>et al.</i> , 2006	HRV	MeanHR, StdNN, RMSSD, LF, HF, LF/HF, TotPow, LFnu and HFnu	600 to 10	Control	54	<ul style="list-style-type: none"> • Cor.: ICC
Munoz <i>et al.</i> , 2015	HRV	SDNN and RMSSD	120, 30, 10	Control	3.387	<ul style="list-style-type: none"> • Cor.: Pearson and Bland–Altman plot on log transformed features • Stat.: Cohen's d
Nardelli <i>et al.</i> , 2017	HRV	SD1 and SD2	60, 25, 15	Rest/Sound	32	<ul style="list-style-type: none"> • Cor.: Spearman correlation and Bland–Altman plot
Nussinovitch <i>et al.</i> , 2011	HRV	MeanNN, StdNN, RMSSD, HRV index, pNN50, LF, HF, TotPow	60 to 10	Control	7	<ul style="list-style-type: none"> • Cor.: ICC

GSR: Galvanic Skin Responses; EMG: Electromyography; EEG: Electroencephalography; BP: Blood Pressure; SC: Skin Conductance; ACC: Actigraphy; Stat: Statistical analysis; Cor: Correlation; ICC: Intra-class correlation analysis.

Table 3.7a: Characteristics of studies investigating ultra-short HRV (cont).

Author, Year	Physiological signals	HRV features investigated	Length (sec)	Condition	N. Sub	Justification for ultra-short HRV adoption
Pandey <i>et al.</i> , 2016	GSR, HRV	MeanNN, StdNN, MeanHR, StdHR, RMSSD, VLF, LF and HF	60	Rest/Stress	15	• None
Papousek <i>et al.</i> , 2010	HRV, BP	MeanHR, LF, HF and LF/HF	180	Rest/Stress	65	• None
Pereira <i>et al.</i> , 2017	HRV	MeanNN, StdNN, RMSSD, pNN20, pNN50, LF, HF, LF/HF, LFnu, SD1, SD2, SampEn and dfa1	220 to 50	Rest/Stress	14	• Stat.: ANOVA between Rest and Stress at different time scale (p<0.05)
Salahuddin <i>et al.</i> , 2007	HRV	MeanNN, RMSSD, pNN50, HRV index, TINN, VLF, LF, HF, LF/HF, LFnu and HFnu	150 to 10	Rest/Stress	24	• Stat.: Kruskal-Wallis test at each condition between 5 min and each time length (p>0.05), and Wilcoxon sign-ranked test between rest and stress at different time length (p<0.05)
Salahuddin <i>et al.</i> , 2007	HRV	MeanNN, RMSSD, pNN50, HRV index, TINN, VLF, LF, HF, LF/HF, LFnu and HFnu	150 to 10	Control	6	• Stat.: Kruskal-Wallis test (p>0.05)
Schroeder <i>et al.</i> , 2004	HRV	MeanNN, StdNN, MeanHR, RMSSD, HF, LF, LFnu, HFnu	360, 180, 10	Control	63	• Cor.: ICC on log transformed features, and multivariate repeated measures
Schubert <i>et al.</i> , 2009	HRV	MeanHR, StdNN, LF, HF, LF/HF and D2	180	Rest/Stress	50	• None
Sun <i>et al.</i> , 2010	HRV, ACC, GSR	MeanNN, StdNN, MeanHR, StdHR, RMSSD, pNN50, LF, HF, LF/HF	60	Rest/Stress	20	• None-referred to literature
Thong <i>et al.</i> , 2003	HRV	SDNN, RMSSD and HF	300 to 10	Control	25	• Stat.: two-way ANOVA (p>0.05),
Wang <i>et al.</i> , 2009	HRV	MeanNN, RMSSD and HF	30	Rest/Stress	735	• None-referred to literature
Wijsman <i>et al.</i> , 2011	HRV, SC, EMG and LF/HF	MeanHR, StdNN, LF, HF and LF/HF	120	Rest/Stress	30	• None
Xu <i>et al.</i> , 2015	GSR, EMG, EEG, HRV	MeanHR, pNN50, LF, HF, LF/HF	180, 30	Rest/Stress	44	• None-referred to literature

GSR: Galvanic Skin Responses; EMG: Electromyography; EEG: Electroencephalography; BP: Blood Pressure; SC: Skin Conductance; ACC: Actigraphy; Stat: Statistical analysis; Cor: Correlation; ICC: Intra-class correlation analysis.

Moreover, among the 29 studies reported, only 7 [140, 148, 149, 159, 164, 166, 167] proposed a model to automatically detect stress using ultra-short HRV features as shown in Table 3.8.

Xu *et al.* [167] enrolled 44 subjects and proposed a model based on K-means clustering and regression analysis to detect stress levels using different physiological signals (i.e., GRS, EMG, EEG and HRV) segmented at 3 min. Wijsman *et al.*

[166] enrolled 40 subjects of which only 18 were used to develop a linear classifier using different physiological signals (i.e., GRS, EMG and HRV) segmented at 2 min. Sun *et al.* [164] enrolled 20 subjects and proposed a tree classifier to detect stress levels using different physiological signals (i.e., GRS, accelerometers and HRV) segmented at 1 min. Choi *et al.* [149] enrolled 3 subjects and proposed a model based on a K-nearest Neighbour Search (IBK) method to detect stress using HRV and principal dynamic mode features segmented at 4 min. Brisinda *et al.* [148] enrolled 113 policemen and proposed a model based on a discriminant analysis classifier to detect stress using HRV features calculated over 1 min and 2 min. Pandey *et al.* [159] enrolled 15 subjects and proposed a stress detection algorithm based on IBK method using 1 min HRV and GRS features. However, Sun *et al.* [164], Choi *et al.* [149] and Brisinda *et al.* [148] proposed models that were built on the assumption that ultra-short HRV features were relevant according to the available literature, although Brisinda *et al.* [148] confirmed their findings using only Intra-class Correlation Analysis (ICC) analysis [172]. Mayya *et al.* [140] enrolled 49 subjects and proposed a method for automatically detecting mental stress using a smartphone and focusing on 1 min HRV features. The model was built on the assumption that ultra-short HRV features were relevant according to the available literature [161], which has been proved to lack a robust method to identify ultra-short HRV features that are good surrogates of short HRV features. Unfortunately, none of these seven papers adopted a rigorous method to select reliable ultra-short HRV features.

Table 3.8: Characteristics of the models aiming to detect stress via ultra-short HRV features.

Author, Year	Models	Physiological signals	Stressors	N. of features	Validation/ Testing	ACC	SEN	SPE
Xu <i>et al.</i> , 2015	Cluster, regression analysis	GSR, EMG, EEG and HRV	PMT	15	LOO	85%	N.R.	N.R.
Pandey <i>et al.</i> , 2016	IBK	GSR, HRV	SCWT, AT, MT and LT	16	N.R.	LSL=67%, MSL =73%, HSL =100%	N.R.	N.R.
Choi <i>et al.</i> , 2009	IBK	HRV	SCWT and AT	2	LOO	77%	N.R.	N.R.
Brisinda <i>et al.</i> , 2014	Discriminant analysis	HRV	Tactical training scenarios	3	N.R.	92%	93%	91%
Mayya <i>et al.</i> , 2015	multinomial logic regression	HRV	SCWT, AT, MT and ST	2	LOO	80.5%	84%	70%
Wijsman <i>et al.</i> , 2011	Fisher's Least Square Linear Classifier	HRV, SC and EMG	AT, MT and LT	9	5-fold cross validation	80%	N.R.	N.R.
Sun <i>et al.</i> , 2010	Decision tree	HRV, ACC and GSR	SCWT and AT	26	10-fold cross fold	92%	N.R.	N.R.

ACC: Accuracy; SEN: Sensitivity; SPE: Specificity; NR: Not Reported; IBK: Nearest Neighbour Search; PMT: Physical-Mental Task; SCWT: Stroop Colour Word Task; AT: Arithmetic Task; MT: Memory Task; LT: Logic Task; ST: Public Speech Task; LOO: Leave One Out cross validation; LSL: Low Stress Level; MSL: Medium Stress Level; HSL: High Stress Level.

3.2.3.3 Conclusion

This short review demonstrated that there is a clear lack of rigorous methods for the selection of good surrogates for short term HRV features. Consequently, there is still the need to develop accurate and valid methods to detect mental stress using ultra-short term HRV analysis in order to enable reliable real-time event detection using wearable sensors and portable devices.

In Chapter 4 a novel approach to assess which ultra-short HRV features are good surrogates of short HRV ones is presented. In Chapter 5, the same framework is applied to identify ultra-short HRV features used to detect mental stress in real-life and laboratory scenarios.

Therefore, the studies presented in the following chapters are the first proposing a rigorous method to detect mental stress using ultra-short HRV analysis.

3.3 Literature review on accidental falls in later-life

The second part of this chapter presents a brief description of the main fall risk factors, existing prevention and prediction programmes and some of the existing monitoring technologies to detect and predict falls in the elderly. Moreover, existing

studies concerning fall prediction and HRV are also discussed (deliverable 2a).

3.3.1 Fall definition

Falls are a serious health problem in the elderly, especially during their permanence in nursing homes and hospitals. In community-dwelling elderly adults, the fall rate is more than 30%; people aged 65 and older have the highest risk of falling, and 50% of people older than 80 years fall at least once a year [173, 174]. The number of elderly adults increases at an accelerating rate and falls represent a costly issue with serious negative consequences for quality of life [173]. In fact, falls are estimated to cost the NHS more than £2.3 Billion per year affecting the family members and carers of people who fall as well, since minor injuries (28%), soft injuries (11%) and fractures (5%) frequently occur following a fall [173]. As well as the physical consequences, there are also psychological ones, constituting what is known as the “post-fall syndrome”, which include fear of another fall, and the loss of self-esteem and independence, compromising the patient's lifestyle and impacting on family caregivers.

Defining a fall is a challenge in itself. For example, the National Database of Nursing Quality Indicators defines a fall as “*an unplanned descent to the floor with or without injuries*” [175], whereas the World Health Organization defines a fall as “*an event which results in a person coming to rest inadvertently on the ground or floor or some lower level*” [176]. Regardless of the definition, a fall is often the result of interactions between intrinsic and the extrinsic risk factors. The former includes patient age, history of recent falls, mobility impairment, urinary incontinence or frequency, certain medications and postural hypertension; the latter refers to physical environments such as rising from a bed or chair, unsupervised toileting and environment hazards [176].

Since falls in the elderly increase morbidity and mortality, unsurprisingly fall prevention has become an important topic either in the development of prevention programmes, including screening tools, or monitoring technologies for the detection and prediction of falls.

However, the prevention, detection and prediction of a fall have often been confused. Therefore, in Table 3.9 the main aim, methods and tools used to prevent, detect and predict a fall are summarised. Falls prevention approaches aim to increase older adults' strength and balance, identify and remove hazards in their environment, increase awareness of falls and associated risk factors, correct clinical conditions that may increase fall risk, or some combination of these approaches in order to reduce the risk of falling. Fall detection aims to mitigate some of the adverse

consequences of a fall (e.g., reduce the time the elderly remain lying on the floor after falling); a fall detection can be identified through an assistive device whose main objective is to alert when a fall event has occurred. Fall prediction aims to predict the next fall in a high-risk subject. At the moment no standardised tests for predicting fall risk have been developed based on inertial sensor-based assessment, but the use of wearable, unobtrusive, low cost and low size, inertial sensors appears promising.

Table 3.9: Fall prevention, detection and prediction.

Tasks	Main aim	Methods	Tools
Prevention	↓ risk of falling	<ul style="list-style-type: none"> • Risk assessment • Intervention • Monitoring 	<ul style="list-style-type: none"> • Scale and physical tests • Physical training, home interventions • High risk referred to local fall clinics/GP
Detection	↓ falls harms	<ul style="list-style-type: none"> • Risk Assessment • Monitoring and automatic warnings 	<ul style="list-style-type: none"> • Ambient sensors • Wearable sensors (mainly accelerometers and gyroscopes)
Prediction	Avoid next fall	<ul style="list-style-type: none"> • Risk assessment • Monitoring • Intervention 	<ul style="list-style-type: none"> • Physical tests • Ambient sensors • Wearable sensors (mainly accelerometers and gyroscopes)

3.3.2 Risk factors

Falls are caused by complex interactions between multiple risk factors (Fig. 3.8), including long-term or short-term predisposing factors, which may be modified by age, disease and the environment. Fall risk factors are commonly referred to by two broad domains [174, 177, 178]:

Intrinsic factors The intrinsic factors (i.e., patient related) are chronic disorders and neurological deficits, increasing age, muscle weakness, gait and balance impairment, postural hypotension, medication use, low body mass index, history of recurrent falls, vision impairment, special toileting needs, urinary incontinence, comorbid illness, depression, and cognitive impairment.

Extrinsic factors The extrinsic factors (i.e., external to the patient) are classified as environmental factors, obstacles in a path of travel, poor lighting, slippery floors, uneven surface, footwear, clothing, and behavioural factors (e.g., activities and choices that can destabilise balance, such as sudden movements or

wearing improper shoes, inappropriate walking aids or assistive devices).

Evidence demonstrates that the risk of falling increases according to the number of intrinsic or extrinsic factors that the subject presents or encounters [174, 177].

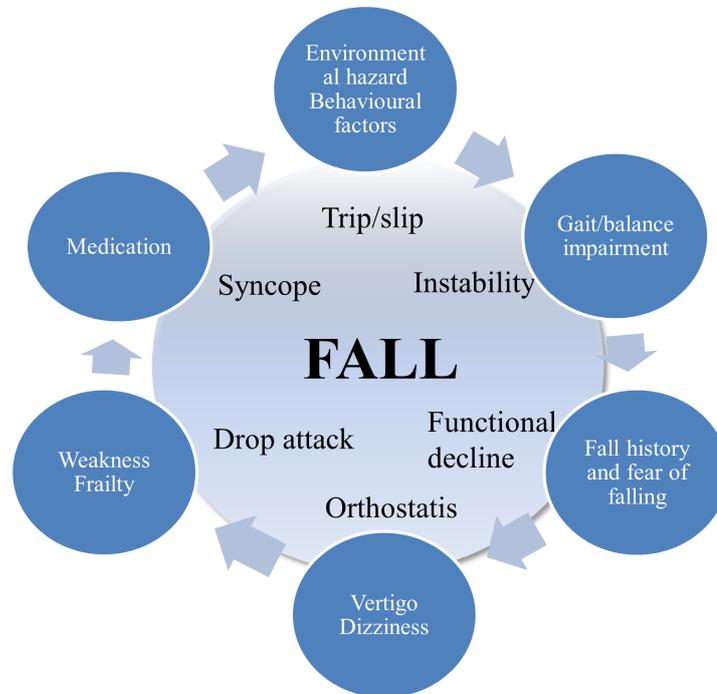


Figure 3.8: Risk factors for falls. The inner circle represents the main causes of a fall, whereas the outer circles are the respective risk factors.

Moreover, it has been demonstrated that women have an increased risk of falling in comparison to men, perhaps due to factors associated with the female gender, like osteoporosis or medication use [177]. Furthermore, individuals with impaired gait have been associated with an increased risk of falls. The elderly with impaired mobility are 1.65 times more likely to fall. Another two risk factors that distinguish between fallers and recurrent fallers are impaired cognition and Parkinson's disease. In particular, some studies showed that Parkinsonism is one of the causes of recurrent falls in the elderly [177]. Finally, poor self-rated health is the remaining significant variable for recurrent falls [177].

The most frequent comorbidities of hospitalised patients for a fall are cardiovascular diseases such as hypertension (63%), coronary atrial fibrillation (30%), artery disease (25%), and congestive heart failure (20%) [63]. Therefore, a primary investigation of health status may be useful in identifying those at risk of falling. In fact, different prevention programmes may be helpful as assessment instruments in

order to identify those at risk of falling [174, 178].

3.3.3 Fall prevention

Since falls can be caused by many factors and the elderly who fall often present several risk factors, an assessment with multiple components that aims to identify a person's risk factor for falling is extremely helpful. All fall prevention programmes include education components, intended to raise awareness about risk factors and interventions that reduce risks, for the purpose of preventing future falls. However, there is no evidence that education alone reduces falls, but it is a vital part of any multifactorial intervention. For instance, a multifactorial intervention with multiple components may be identified as the first step to address the risk factors for falling [174, 176].

3.3.3.1 Prevention tools

The multifactorial assessment followed by multifactorial intervention is believed to be necessary for fall prevention [174, 176, 178]. Older people who present medical attention because of a fall, or recurrent falls or gait problems are often offered a multifactorial assessment, performed by healthcare professionals. The multifactorial assessment may include the following:

- identification of falls history;
- assessment of gait;
- assessment of osteoporosis risk;
- assessment of fear relating to falling;
- assessment of visual impairment;
- assessment of cognitive impairment and neurological examination;
- assessment of urinary incontinence;
- cardiovascular examination and medication review.

On the other hand, all older people with recurrent falls or assessed as being at elevated risk of falling are considered for an individualised multifactorial intervention, which may include:

- strength and balance training;

- home hazard assessment and intervention;
- vision assessment and referral;
- physical restraints;
- medication review with modification or withdrawal.

Methods and tools for assessing fall risk in home-dwelling older persons with minor functional problems are several, for instance functional balance and mobility assessment, including the use of the Berg Balance Scale [179, 180], the Tinetti Mobility Scale [181], the Morse fall scale [182], the Functional Gait Assessment [183], the Balance Evaluation Systems Test [184], the STRATIFY fall scale [185], the Hendrich II Fall Risk Model [186] the Timed Up-and-Go [187], the 5-step test and floor transfer [188], the functional reach [189], getting up from lying on the floor [190], one-leg balance [191], stop walking when talking [192] and timed walk/distance walked [188].

However, most of these tools that have been proposed for fall risk assessment have discriminated poorly between fallers and non-fallers, none of which are universally accepted. In fact, a recent systematic review [193] reported that the accuracy of tools for detecting fall risks in acutely hospitalised ill-patients such as the Morse, STRATIFY and Hendrich II Fall Risk Model tools are very low. The evidence base indicates that multifactorial interventions are effective at reducing falls but not able to detect or predict a fall [176]. A recent Cochrane review showed that multifactorial interventions in hospitals reduce the rate of falls (rate ratio 0.69, 95% CI: 0.49 - 0.96), but it was not easy to isolate their specific effect in predicting a fall [194].

Overall, these preventing programmes may only help to promote independence and improve physical and physiological functions.

3.3.4 Fall prediction tools

Some fall risk assessment tools are available with some evidence to support their use in predicting the risk of falls. Physical and mobility tests are sometimes used to predict falls. However, the sensitivity and specificity of these tools are low. In fact, a recent systematic review concluded that one of the most used physical tests: the Timed Up and Go test, commonly used as screening tool to assist clinicians to identify patients at risk of falling, has also been used to predict falls [195]. However, the analysis carried out by Barry *et al.* [195] indicated that the Timed Up and Go score is not a significant predictor of falls with a pooled specificity of 74% (95% CI 0.52-0.88) and sensitivity of only 31% (95% CI 0.13-0.57).

Another study, conducted by Tiedemann *et al.* [196] compared the abilities of eight mobility tests (i.e., sit-to-stand test [197], pick-up-weight test [198], half-turn test [198], alternate-step test [198], six-metre-walk [199], stair ascent and descent [200], falls surveillance [201]) for predicting multiple falls in a large sample of older community-dwelling people. The tests demonstrated poor to fair sensitivity and specificity (Table 3.10) in identifying older people at risk of multiple falls. Therefore, these tests should only be used as initial screens for identifying older people in need of further assessment.

Table 3.10: Mobility tests proposed in literature for predicting falls.

Tasks	Author, year	Sensitivity	Specificity
Timed Up and Go	Barry <i>et al.</i> , 2014*	31%	74%
Sit-to-stand once	Tiedemann <i>et al.</i> , 2008	49%	58%
Sit-to-stand five times	Tiedemann <i>et al.</i> , 2008	66%	55%
Pick-up-weight test	Tiedemann <i>et al.</i> , 2008	11%	93%
Half-turn test	Tiedemann <i>et al.</i> , 2008	78%	28%
Alternate-step test	Tiedemann <i>et al.</i> , 2008	69%	56%
Six-metre walk	Tiedemann <i>et al.</i> , 2008	50%	68%
Stair ascent	Tiedemann <i>et al.</i> , 2008	54%	58%
Stair descent	Tiedemann <i>et al.</i> , 2008	63%	55%

*Systematic review with meta-analysis

3.3.5 Monitoring technologies for fall detection and prediction

Several fall risk prevention tools have been developed to identify at-risk populations and guide intervention by highlighting remediable risk factors for falls and fall-related injuries. Despite the numerous clinical scores developed, these methods often depend on individual observation and subjective interpretation, which make the assessment resulting inconsistent [202] and with limited accuracy in recall [203]. Some standard tests also require subjective judgements. The need for objective and clinically applicable methods is clear. Therefore, modern sensor technologies and healthcare can help to close this gap and allow for un-obtrusive quantitative monitoring of patients in their environment.

There is an increasing interest in an alternative emerging strategy as well as in the development of sensor systems with light and sound alarms used to predict and not just to prevent a fall [204]. The problem of these systems is that in all the applications reported in the literature, the alarm is generated every time the elderly are exposed to a risky situation (i.e., rising from a bed) regardless of the real condition of the subject. This generates an unsustainable rate of false positives,

causing the abandonment of the technologies.

According to the existing literature [173], there are two main classes of sensor technologies in terms of sensor position: wearable and non-wearable. However, most of the current technologies are based on the detection of movement by pressure, position and infrared light, which may be used for the detection and not the prediction of the fall. In fact, few sensor technologies are used to predict falls in the elderly, which are based on the study of how biomedical signals vary before a fall happens.

Fall detection and prediction systems are both aimed at reducing the consequences of a fall using various sensors and algorithms, however, there are some key differences.

Fall detection systems alert the user and healthcare provider after a fall has occurred to expedite and improve the medical care provided. These systems are aimed at identifying different kinds of falls: falls from walking or standing, falls from standing on supports, falls from sleeping or lying in a bed and falls from sitting on a chair. These systems often use threshold-based algorithms to detect falls. The performance metrics for fall detection systems include precision (true positive rate), specificity (true negative rate), and the false positive rate. Moreover, fall detection systems mainly focus on physiological risk factors such as gait, mobility, and vision.

On the other hand, fall prediction systems are aimed at alerting the subjects before the occurrence of a fall thus preventing the emotional and health consequences of a fall. These systems should identify all scenarios and circumstances leading to falls and provide a framework to predict them. This framework must be constructed based on data acquired from various scenarios surrounding fall-related events. Information on fall-related events is usually collected through questionnaires, fall diaries, and phone calls. Although these data collection practices do provide relevant information, the information is not always reliable. This happens because people often forget or remember incorrectly the exact conditions of their fall [205]. Therefore, this information should often be augmented with physiological data collected from various sensors to improve the overall reliability and accuracy of the proposed systems.

3.3.5.1 Non-wearable sensors

Most non-wearable sensors have been developed to detect falls and these devices can be gathered in three categories [173, 206]:

1. video-based monitoring of real time movement;
2. acoustic frequency or floor vibration-based monitoring, where falls are detected

by analysing the frequency components of sound or vibration caused by the impact of the falls;

3. activity monitoring devices.

As far as the first two categories are concerned, they are primarily focused on the detection of falls; in fact, the intent of these two categories is only to minimise the time between a fall and arrival of medical attention [173, 206–208].

As far as the third category is concerned, there is a wide array of passive monitoring activity but also some intelligent sensors such as Intelligent bed care system, which combines bed sensors with a recording of physiological measurements [209]. Spetz *et al.* [209] developed the first prototype of a sensor technology to predict a fall developing complex algorithms which were able to identify potential problems measuring heart rate and respiration of an older person. Therefore, if an older person's heart rate and respiration raised beyond his/her expected physiological range the patient is considered at risk and an alarm call for medical attention is made [209]. Although falls were reduced with this system [209], the authors did not report the level of significance, which fails to draw any firm conclusions. However, it is well-established that falls do not occur only in small areas such as bed and chair but in a variety of conditions and a wide array of spatial and temporal distributions, therefore, these technologies may be restrictive [173].

3.3.5.2 Wearable sensors

Wearable sensors are mobile electronic devices that can be worn or embedded in clothing or accessories [206]. Recent interest in this field has produced a variety of applications related to the elderly. Most of them can be attached to a patient's thigh or foot and they are mostly based on recognising the change in pressure and proper acceleration [210–213]. In fact, most of the studies in the literature use accelerometry data along with threshold-based algorithms to detect fall-related events [205].

Several challenges still exist in the implementation of this technology; in fact, it appears that the sensors based on pressure and proper acceleration are not feasible and accurate due to the number of false alarms, which could be nuisance for older people and could lead to disuse [173, 206].

On the other hand, some devices show promising developments in the prediction of falls through the acquisition of physiological measurements, although some of them are not available yet commercially or are being used in limited settings [206]. However, it is worth mentioning some of them such as the MEMSWear [214, 215], AMON [216] and Smart Vest [217] devices.

MEMSWEAR is a wearable smart shirt that can predict fall events through the use of motion sensors such as gyroscopes and accelerometers, and physiological sensors used to analyse ECG signals and blood pressure. These measurements are important data inputs to complex mathematical models used to predict the imminence of a fall and alert medical attention. As far as the acquired biomedical signals, the blood pressure sensing is the first physiological sign that is investigated to determine abnormality in humans that might bring about fainting. ECG sensing is the next biomedical signals to determine an abnormality. An embedded microcontroller is used to calculate heart rate and monitor the ECG signal for life-threatening arrhythmias. This technology is able to remotely monitor vital human signals, which when implemented in a fall prediction model may be a reliable support for fall prediction; however, there have been no clinical trials to confirm the validity of the technology.

AMON is a wrist-worn medical monitoring and alert system, which collects and evaluates vital signs such as heart rate and heart rhythm via ECG, temperature, blood pressure and blood oxygen saturation. Heart rate, skin temperature, oxygen saturation are monitored continuously, but blood pressure and one-lead ECG are only measured 3 times a day. However, some inconsistency in the vital data have been reported, in fact, although the blood pressure was considered reliable, body temperature could not be accurately analysed, blood oxygen saturation was not a reliable signal and the ECG recording provided a high level of noise.

Finally, Smart Vest is a wearable physiological monitor system, which is built into a t-shirt. Different biomedical signals such as heart rhythm via ECG, heart rate, oxygen saturation, body temperature, blood pressure and galvanic skin response are acquired. However, Smart Vest is still a prototype and many issues need to be addressed [217].

One of the main limitations of wearable sensors based on physiological monitoring to predict falls is the number of false positives in machine learning algorithms due to the rarity of the events. In fact, the evaluation of fall detection and prediction approaches has almost exclusively focused on the accuracy of the detection or prediction algorithm. There is, therefore, the need to develop new classification methods for improving both sensitivity and specificity [218].

3.3.6 HRV and fall prediction

Many existing wearable sensors are based on gyroscopes and accelerometer signals and very few studies have investigated HRV as a tool to assess the risk of falling. However, it has been demonstrated that there is a significant association between a

depressed HRV and the risk of falling, suggesting that a depressed HRV could be a new independent risk factor for falls [63, 219, 220]. Nevertheless, the majority of studies have focused their attention on fall detection rather than fall prediction via HRV [221–223].

To the best of the author's knowledge, only two studies [64, 224] investigated the discrimination power of HRV features for fall prediction. Isik *et al.* [224] conducted a retrospective study on 33 older adults who had fallen in the last 12 months and the control group included 31 subjects who had never experienced falls. The patients in the study group were examined with 24h Holter ECG. HRV and Heart Rate Turbulence (HRT) features were extracted and they demonstrated that older subjects with recent falls had significantly worse HRT parameters than matched non-falling counterparts. In Table 3.11, only the HRV features investigated and the respective p-values are reported. However, no significant changes were observed in the HRV features' variations between fallers and non-fallers.

Table 3.11: HRV features from 24h in fallers and non-fallers [224].

HRV Features	Fallers ($n = 33$)	No-fallers ($n = 31$)	<i>P</i>-value
HR (1/min)	72.65 ± 10.62	75.29 ± 6.67	0.243
StdNN (ms)	111.18 ± 34.36	124.78 ± 32.61	0.110
RMSSD (ms)	29.70 ± 19.51	27.28 ± 17.65	0.606
PNN50 (%)	4.89 ± 7.27	5.15 ± 7.75	0.888
24 hour LFnu	52.7 ± 15.2	54.6 ± 17.4	0.650
24 hour HFnu	24.4 ± 11.6	22.2 ± 10.1	0.417
LF/HF ratio	3.0 ± 2.2	3.3 ± 2.1	0.490

Melillo *et al.* [64] did not report any values for the HRV features investigated between fallers and non-fallers, whereas they developed a model to predict falls using HRV features extracted from 24h ECG recordings. The developed model was able to automatically predict a future fall among hypertensive patients with 72% accuracy as shown in Table 3.12. However, the standard classification methods used in [64] reported high number of false positives as shown in the low sensitivity values.

Table 3.12: Best performance of the adopted classification methods using HRV features from 24h in fallers and non-fallers [64].

Methods	AUC	ACC	SEN	SPE	DOR (95% CI)
RF	46.3%	67.3 %	21.3 %	85.1 %	1.5 (0.6–3.6)
RTF	51.5%	67.9 %	21.3 %	86.0 %	1.6 (0.7–3.9)
AB	51.7%	68.5 %	25.5 %	85.1 %	2.0 (0.9–4.5)
MB	54.1%	63.7 %	17.0 %	81.8 %	0.9 (0.4–2.2)
RB	63.9%	69.0 %	40.4 %	80.2 %	2.7 (1.3–5.7)
RB and PCA	67.6%	72.0 %	51.1 %	80.2 %	4.2 (2.0–8.7)

RF: Random Forest; RTF: Rotation Forest; AB, AdaBoost; MB, MultiBoost; RB, RUSBoost; RB and PCA, RB enhanced with PCA. AUC: Area Under the Curve; ACC: Accuracy; SEN: Sensitivity; SPE: Specificity; DOR: Diagnostic Odds Ratio.

3.3.7 Conclusion and limitations

Overall, there are many preventive programmes promoting independence and improving physical and physiological functions, however, the evidence base indicates that multifactorial interventions are effective at reducing falls, but not able to predict a fall [176]. Moreover, these methods often depend on individual observations and subjective interpretation, which make the assessment inconsistent [202] and with limited accuracy. Therefore, the need for objective and clinically applicable methods is clear.

Along with the prevention of a fall, more important and effective is the prediction of a fall before an injury may aggravate the elderly situation and enhance the mean cost of a fall. Therefore, the current interest has been shifted from preventing a fall to “fall prediction” and this is the main reason for a growing number of wearable technologies released on the market aiming to predict falls. Nevertheless, there are still challenges and many researchers are still wondering whether the current sensor systems can reliably predict falls and fall-related injuries in older people.

Many sensor technologies are in their early stages of development and they need extensive testing regarding validity, reliability, acceptability and utility. In fact, they present several limitations regarding the reliability of physiological parameters as well as accurate algorithms for fall prediction [173, 206].

A recent systematic review [218] highlighted that many of the proposed technologies presented several limitations including an elevated occurrence of false alarms, the obtrusiveness of those technologies and their cost-effectiveness [64]. Regarding cost-effectiveness, the majority of the proposed approaches require the use of additive sensors (mainly accelerometers, gyroscopes or ambient sensors) having no other

direct utility for older citizens' health and therefore, determining unsustainable additional costs [64, 218]. Also, the mechanism that accelerometers, gyroscopes or ambient sensors use, cannot detect all the risk factors for falls and even the sensors to acquire those signals are not comfortable to wear for the elderly.

Therefore, recent studies have investigated the hypothesis that physiological variation in biomedical signals (i.e., HRV) can be a potential marker of fall risk and prediction. In fact, it has been demonstrated that there is a significant association between a depressed HRV and the risk of falling, suggesting that a depressed HRV could be a new independent risk factor for falls [63, 219, 220]. However, the majority of studies have focused their attention on fall detection rather than fall prediction via HRV [221–223].

3.4 Conclusions

This chapter identified gaps in the existing literature on stress detection and fall prediction via HRV.

The review of the existing literature on mental stress and HRV identified multiples gaps proving that future studies are needed to confirm the behaviour of ultra-short HRV features during stress. Furthermore, there is an evident lack of rigorous methods for the selection of ultra-short HRV features that are good surrogates of short HRV features and consequently, there is still the need for developing accurate and valid methods to detect mental stress using ultra-short term HRV analysis in order to enable reliable real-time event detection using wearable sensors and portable devices. In fact, in Chapter 4 a novel method to assess which ultra-short HRV features are good surrogates of short ones is presented and different experiments were designed and carried out to demonstrate its efficacy (Chapter 5).

The current evidence of “prediction of falls in the elderly” is inconsistent whether the multifactorial interventions tools and current sensor systems can predict falls and fall-related injuries in the elderly. However, recent studies confirmed that multifactorial interventions are effective at reducing falls, but are not able to predict a fall whereas wearable sensors still lack validity, reliability, acceptability and utility. Therefore, objective and clinically applicable methods are necessary.

Indeed, one of the main limitations of wearable sensors based on physiological monitoring to predict falls is the number of false positives in machine learning algorithms due to the rarity of the events. In fact, the evaluation of fall detection and prediction approaches have almost exclusively focused on the accuracy of the

detection or prediction algorithm. There is, therefore, huge interest in developing more structured classification approaches for improving both algorithms' sensitivity and specificity rates [218]. Moreover, although it has been demonstrated that there is a significant association between a depressed HRV and the risk of falling, few studies have investigated HRV as a biomarker to predict falls. As matter of fact, HRV could be a non-invasive and a cost-effective tool for fall prediction, which could show better results than current prediction programmes. Therefore, there is the need for methods based on non-invasive biomedical signal (e.g., HRV) analysis to automatically identify future fallers and reduce the number of false positives.

A robust approach to reduce false positive numbers is presented in Chapter 4 and novel results are presented in Chapter 6.

In conclusion, the review of the existing literature on mental stress and HRV identified multiples gaps proving that:

- future studies are needed to confirm the behaviour of ultra-short HRV features during stress;
- there is a clear lack of rigorous methods for the selection of ultra-short HRV features that are good surrogates of short HRV features.

Likewise, the review of the existing fall prevention, detection and predictions tools highlighted different gaps:

- the need for objective and clinically applicable methods to prevent and predict falls;
- the need to develop structured classification approaches for improving both the algorithms' sensitivity and specificity to predict a fall.

The following chapters present the solutions to the identified gaps, proposing new approaches (Chapter 4) and applying them to two specific case studies (Chapters 5 and 6).

Chapter 4

Development of Methods and Tools to Monitor Cardiovascular and Autonomic Response in Real-life Settings

4.1 Chapter overview

The previous chapter reviewed the existing literature on mental stress detection and fall prediction in later-life. Methodological and problem-specific gaps were identified for both case studies.

This chapter presents the frameworks and tools developed to answer the research questions (refer to Chapter 1, section 1.4) arising from the theoretical limitations outlined in the current literature in monitoring the CVS and the ANS interaction in real-life settings and from the existing gaps identified for the specific case studies.

The main objectives (Obj 1, 2 and 3) are tackled in sections 4.2.1 and 4.2.2.

4.2 Theoretical approaches to biomedical signal processing and machine learning in real-life settings

Signal processing and machine learning techniques are frequently combined to solve specific problems. In particular, signal processing is used to pre-process data coming from wearable sensors before applying machine learning techniques to detect or predict adverse healthcare events. Proper use of both signal processing and machine

learning techniques is the key to successfully detect or predict dysfunctions between the CVS and the ANS. However, recent studies show inappropriate use of either signal processing or machine learning techniques resulting in unreliable algorithms for the detection or prediction of adverse events in real-life settings. Therefore, the approaches and tools described in this chapter could accelerate the development of sensor processing systems by providing useful tools for the advancements in applied signal processing and machine learning techniques to real-life data.

The first part of this chapter is focused on a novel framework to investigate to what extent biomedical signals (i.e., HRV) can be shortened (i.e., $< 5\text{min}$) without losing important physiological information. In other words, whether HRV features extracted from ultra-short excerpts (i.e., $< 5\text{min}$) can be considered good surrogates of short HRV features. In fact, this is the most prominent requirement for wearable sensors to detect or predict adverse healthcare events in quasi-real-time.

The second part of this chapter is focused on defining frameworks to improve machine learning techniques in order for them to cope with small and unbalanced datasets. In fact, another important issue for establishing a reliable supervised learning strategy in real-life settings and preventing over-fitting problems is to properly make use of the available samples, especially when the number of available samples is small or when one or more classes occur far less frequently than others.

4.2.1 How to determine surrogates of biomedical signals

Many applications claim to perform real-time problem detection using only a few seconds of physiological signals (e.g., HRV, conductance skin response, breathing rate and so on). In particular, ultra-short HRV (less than 5 minutes) is being used more and more to investigate CVS and ANS dysfunction using wearable sensors in real-life settings. Therefore, the demands of ultra-short term HRV analysis for monitoring individual's well-being status is significantly increasing due to the diffusion of wearable sensors in the healthcare industry, especially for its usefulness in mobile phones and smartwatches. In e-health monitoring, in fact, the conventional 5 minutes recordings might be unsuitable, due to real-time requirements. Nevertheless, numerous challenges have arisen by shortening HRV excerpts below 5 minutes.

In the existing literature (refer to Chapter 3, section 3.2.3), there are not yet clear guidelines on how to analyse HRV in the ultra-short term and there are no clear frameworks to identify reliable subsets of ultra-short HRV features for the automatic detection of adverse healthcare events. Consequently, there is still the need to develop accurate and valid methods to detect adverse events using ultra-short term HRV analysis, in order to enable reliable real-time adverse event detection

using wearable sensors and portable devices.

The proposed frameworks aim to explore to what extent ultra-short HRV features can be used to estimate short term ones, which are still to be considered as a benchmark for HRV analysis.

Due to the lack of rigorous references for ultra-short HRV analysis, the frameworks presented in the next section are in alignment with the medical literature on surrogate outcomes [143, 144]. In fact, in medicine, and particularly in clinical trial design, in order to cope with this kind of problem, the concept of a surrogate endpoint (or marker) was introduced.

Definition of surrogates In medicine, various definitions of a surrogate have been proposed over the years. A surrogate measure is a marker, which is used to estimate a real clinical endpoint, when this is undesired (e.g., death) or when it cannot be directly observed or measured. Several regulatory bodies (e.g., the FDA¹, NICE²) have started to accept evidence from clinical trials that show a direct clinical benefit in using surrogate markers.

As defined by Temple [225], *“a surrogate endpoint is a laboratory measurement or a physical sign used as a substitute for a clinically meaningful endpoint that measures directly how a patient feels, functions, or survives. Changes induced by a therapy on a surrogate endpoint are expected to reflect changes in a clinically meaningful endpoint”*.

In a workshop organized by the NIH [226], the following definition was also recommended: *“a biomarker intended to substitute for a clinical endpoint. A clinical investigator uses epidemiologic, therapeutic, pathophysiologic, or other scientific evidence to select a surrogate endpoint that is expected to predict clinical benefit, harm, or lack of benefit or harm”*.

In medical practice, different statistical approaches have been developed in an attempt to validate surrogate endpoints [144]. These approaches have both strengths and limitations. Most importantly, different approaches may yield different results.

Existing methods to find surrogates in medicine Various statistical approaches have been proposed, however, it appears that all validation methods focus on the following three requirements [143, 144]:

1. a valid surrogate must be correlated with the clinical endpoint (i.e., with a correlation coefficient above 0.7 as specified in [227]);

¹Food and Drug Administration, <https://www.fda.gov/>.

²National Institute for Health and Care Excellence, <https://www.nice.org.uk/>.

2. a valid surrogate should capture a reliable and sufficiently large portion of the treatment effect on the clinical endpoint;
3. a valid surrogate should be able to estimate the treatment effect on the clinical endpoint.

Validation of these three requirements needs distinct statistical approaches.

In fact, proving whether or not a marker is a good surrogate of a real clinical outcome can be quite difficult, and a combination of appropriate statistical and correlation tests is required. Although a rich body of literature has been produced to answer this question, some authors still disagree on the issue and the sentence “*a correlate does not make a surrogate*”, first used by Fleming *et al.* [143], became a mantra in this field. In fact, there is a common misconception that if a marker correlates with the true clinical outcome, it can be used as a valid surrogate endpoint, replacing the true clinical outcome. However, a much stronger condition than correlation is required to be sure that a surrogate is valid and can be used to replace a real clinical outcome. Another common misconception is that a marker X can be considered a good surrogate of a clinical outcome Y, if statistical null-hypothesis tests demonstrate no-significant differences between X and Y. This is a major misconception because statistical differences may reveal themselves only in particular conditions (e.g., when a sufficient number of measures are observed). In addition, both correlation and statistical tests are often used improperly (e.g., parametric tests used for non-normally distributed features).

Based on the definitions reported above, in biomedical engineering, we can define a surrogate as “*a physiological signal used as a substitute of a physiologically justified signal that can reflect changes in the subject's status reliably compared to the gold standard*”.

Therefore, in order to identify ultra-short HRV features that are “good” surrogates of the gold standard (5 min HRV features) novel frameworks are developed following the main requirements used in medicine. The presented frameworks are developed for both assessing the validity of ultra-short HRV features in a control situation and identifying reliable subsets of ultra-short HRV features to allow the detection of an adverse healthcare event. The former case aims to prove that ultra-short HRV features do not change behaviour with respect to short term ones. The latter aims to assess that:

1. ultra-short HRV features behave as short term ones for the same conditions (i.e., at rest or during stress), intra-group assessment;

2. ultra-short HRV features maintain the same behaviours for the two conditions over different lengths, inter-group assessment.

4.2.1.1 Framework to assess the validity of ultra-short HRV features in a control condition

A general framework in agreement with medical practice [228] and the existing literature is proposed by the author in Fig. 4.1, in the case where HRV features are investigated only during a control condition in segments shorter than 5 min.

The only use of statistical or correlation tests to explore whether ultra-short HRV features can be considered good surrogates of short term ones, is methodologically erroneous. In particular, no conclusion can be drawn relying only on statistical tests proving that the p-value between ultra-short HRV features and the benchmark (short term HRV features) is greater than 0.05. Indeed, a p-value greater than 0.05 is not a statistical discriminator. For instance, a p-value could be greater than 0.05 only because the number of subjects enrolled is not sufficient. Many studies in the literature (refer to Chapter 3, section 3.2.3), concluded that ultra-short HRV features were good surrogates of short term ones, if no-significant differences were observed, using a significance threshold greater than 0.05 (p-value>0.05). Unfortunately, this result is debatable because, while a p-value<0.05 is conventionally used to support the hypothesis that two distributions are significantly different, it is well-known that no conclusions can be drawn for p-value greater than 0.05, as detailed in [169]. Alternatively, as shown in Fig. 4.1, before performing proper statistical tests, it should be considered whether the features are significantly correlated over different time scales. Significant correlation suggests that there is a significant association. Nonetheless, this association could be biased. The Bland-Altman method [170] estimates this bias and how it diverges with an increase in the magnitude of the short term feature (i.e., benchmark). According to this test, two features are considered unbiased, if the dispersion of their mean difference remains within a conventional threshold (i.e., 95% LoA, Line of Agreement) [130]. Once correlation has been proven and bias excluded, the statistical significance can be explored. Munoz *et al.* [156] proposed the use of the Cohen's d statistic to quantify the agreement of HRV features over different time scales relative to their within-group variation [229]. Therefore, according to the proposed algorithm, a feature can be considered a good surrogate if correlated, non-biased and significantly in agreement across the different time scales.

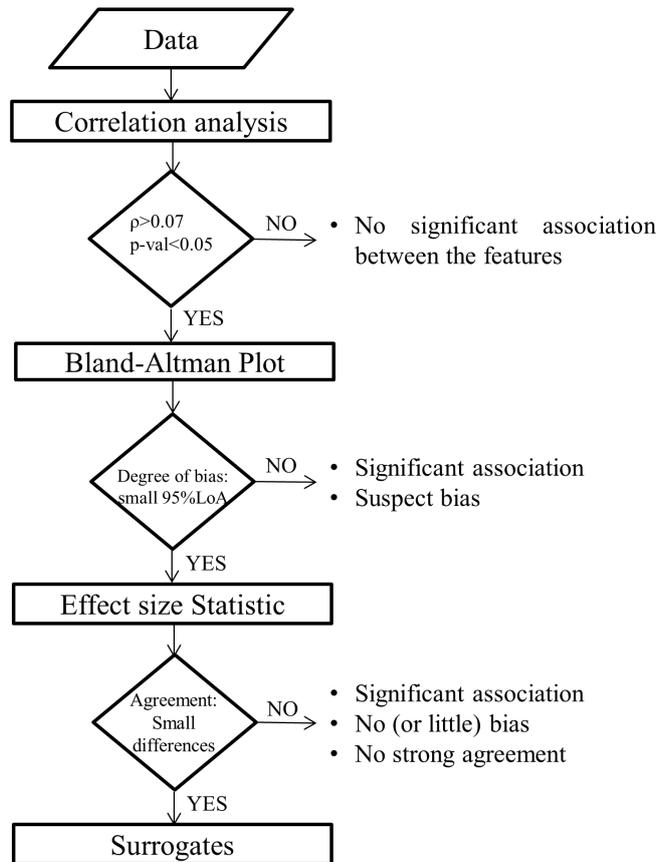


Figure 4.1: The algorithm to assess if ultra-short HRV features can be considered good surrogates of short-term ones for one condition (e.g., only at rest). The correlation coefficient is represented as ρ ; p-val is the p-value associated with correlation analysis; LoA is the line of agreement in Bland-Altman plot.

The algorithm reported in Fig. 4.1 can be further articulated in the case where the ultra-short HRV features are non-normally distributed (Fig. 4.2). As far as correlation tests are concerned, there are several non-parametric tests, which have been proposed. Alternatively, HRV features can be log-transformed before using a parametric test. The Bland-Altman test is parametric too, as it calculates the 95% LoA around the mean. In the case of non-normally distributed features, the dispersion should be investigated around the median, and not the mean, when computing the 95% LoA [170, 230]. Finally, since Cohen's d statistic assumes a normal distribution for the input features, a log-transformation of HRV features is required before applying this test [171]. Alternatively, the Cliff's Delta statistic could be used for non-normally distributed data as it is a non-parametric effect size measure that quantifies the amount of difference between two groups of observations

beyond p-value interpretation [231].

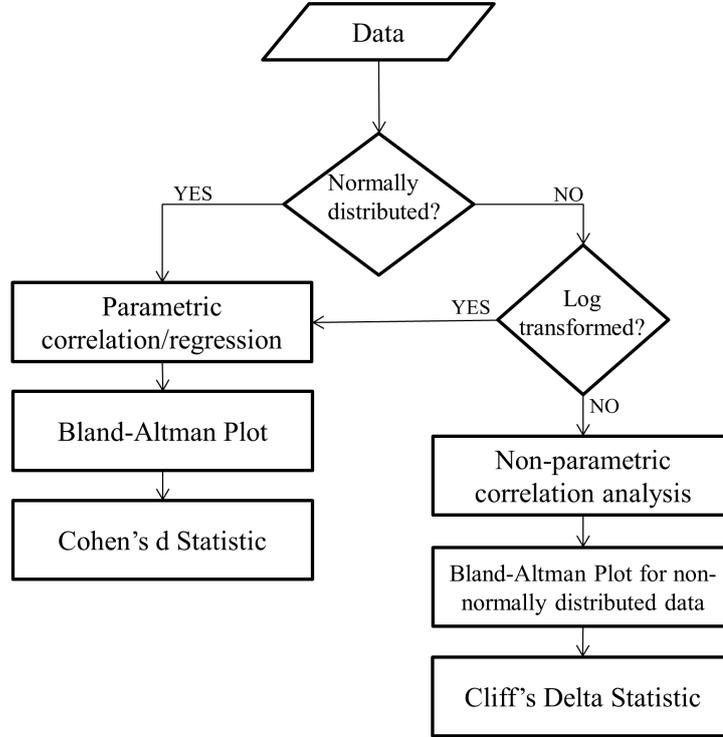


Figure 4.2: Methodological framework to assess the validity of ultra-short HRV features as surrogates of 5 min HRV ones under a control condition. All the analyses should be run between the benchmark and each time scale.

4.2.1.2 Framework to assess the validity of ultra-short HRV features under two conditions

A novel framework in alignment with the best available medical practice is presented in Fig. 4.3 to investigate ultra-short HRV features under two different conditions (e.g., rest and stress). The full methodological framework is also presented in Chapter 5, section 5.2.6, Fig. 5.7.

In the case where two different conditions are explored (e.g., stress VS rest), further adjustments are required to the previous framework (Fig. 4.2). In fact, the algorithm proposed in Fig. 4.3 aims to prove that:

- ultra-short HRV features behave as short term ones for the same conditions (i.e., at rest or during stress), intra-group assessment;
- ultra-short HRV features maintain the same behaviours for the two conditions over different lengths (i.e., if StdNN diminishes during stress, this change

should be observed both for short and ultra-short term data), inter-group assessment.

As the first step, surrogate features have to be correlated with benchmark ones (i.e., short term HRV) both for a control condition (e.g., a rest phase) and during the event to be detected (e.g., a stress phase). This can be verified using intra-group correlation analysis for different time lengths, i.e., for the same condition. For instance, StdNN (as well as any other HRV feature) extracted from 5min excerpts during rest (or stress), has to be significantly correlated with StdNN extracted from any shorter than 5min excerpts during rest (or stress).

For the second step, visual investigation of bias between means (or medians for non-normally distributed features) has to be performed via Bland Altman plots for each condition (e.g., rest and stress).

For the third step, the set of surrogate features has to preserve a large proportion of the information of the event to be detected (i.e., a significance test for each time scale and/or trend analysis). This can be verified using inter-group statistical tests for each time length, but for the different conditions. Therefore, it is necessary to verify, using non-parametric tests (unless HRV features are normally distributed or log-transformed), which ultra-short HRV feature maintains statistical evidence that the median (or mean) differs significantly ($p\text{-value} < 0.05$) for the two different conditions (e.g., between rest and stress) across the time period windows.

For the fourth and last step, the trends of the HRV features (i.e., if HRV features decrease or increase during stress) should remain consistent across time lengths. In fact, an HRV feature can be assumed to maintain the same behaviour across different time period windows if the statistical significance test has a p-value less than 0.05 between the control and the experimental conditions for each time length and if the ultra-short HRV feature's trends change between the control and the experimental conditions consistently with the equivalent short HRV feature (e.g., if MeanNN decreases significantly during stress over 5min, this significant trend has to be consistently maintained for shorter time lengths).

Once those 4 steps have been performed, it can be assumed that an ultra-short HRV feature is a good surrogate of the equivalent short one, if:

1. the feature maintains the same behaviour between control and experimental conditions over the investigated time period windows;
2. the ultra-short HRV feature was highly and significantly correlated (e.g., a correlation coefficient greater than a given threshold (e.g., 0.7) and p-value

lower than 0.05), with the corresponding short feature, across the time period windows under both control and experimental conditions.

Also, after having identified the subset of good surrogates, their discrimination power in detecting the event of interest with sufficient accuracy can be explored in order to automatically classify the two conditions.

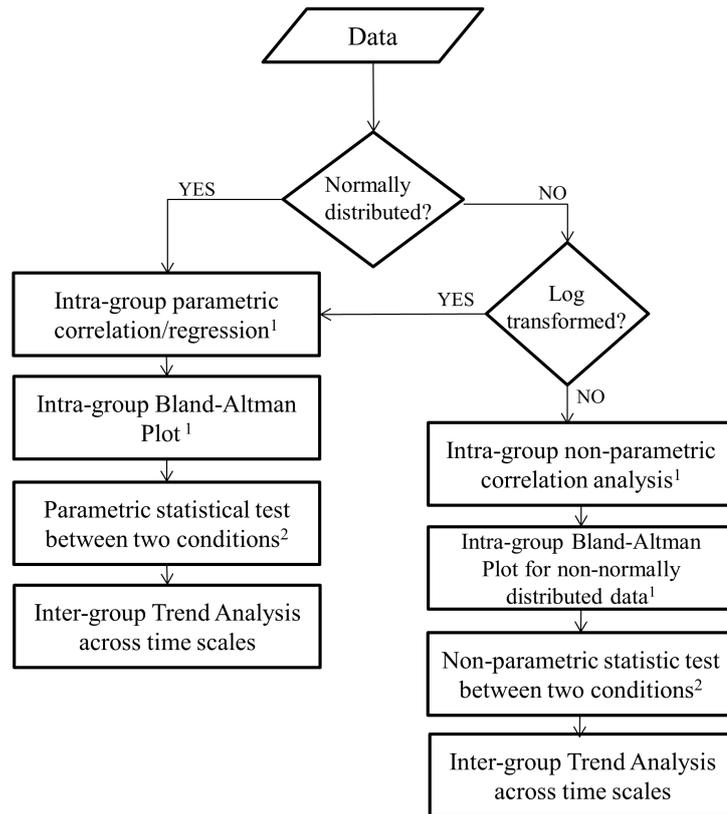


Figure 4.3: Methodological framework to assess the validity of ultra-short HRV features as surrogates of 5 min HRV ones to detect an adverse event.

¹ The analysis should be run between the benchmark and each time scale during both control and experimental conditions. ²Repeated at each time scale.

According to these requirements a Matlab tool to determine ultra-short HRV features as good surrogates of short HRV features under one or two conditions was developed and is reported in Appendix A. The developed tool is generalised for any application requiring the use of HRV analysis in the ultra-short term. Fig. 4.4 shows the pseudocode for the proposed frameworks.

```

Input file/files
If [only one condition]
  if [normally distributed]
    then parametric correlation/regression between the benchmark and each time scale
    then Bland-Altman Plots between the benchmark and each time scale
    then Cohen's statistic between the benchmark and each time scale
  else [non-normally distributed]
    then non-parametric correlation between the benchmark and each time scale
    then Bland-Altman Plots for non-normally distributed data between the benchmark and each time scale
    then Cliff's Delta statistic between the benchmark and each time scale
  end
else [two conditions]
  if [normally distributed]
    for [each condition]
      then parametric correlation/regression between the benchmark and each time scale
      then Bland-Altman Plots between the benchmark and each time scale
    end
    for [each time scale]
      then parametric tests between the two conditions
      then trend analysis across time scales
    end
  else [non-normally distributed]
    for [each condition]
      then non-parametric correlation between the benchmark and each time scale
      then Bland-Altman Plots non-normally distributed data between the benchmark and each time scale
    end
    for [each time scale]
      then non-parametric tests between the two conditions
      then trend analysis across time scales
    end
  end
end

```

Figure 4.4: Pseudocode to assess the validity of surrogates.

The developed tool takes as input a matrix of HRV features extracted from the benchmark length and matrices of HRV features extracted over different time scales. The input matrix is a numerical matrix presenting observations arranged in rows and features in columns as shown in Fig. 4.5. The dimension of the input matrices depends on the number of observations and features extracted.

	1	2	3	4	5	6	7	8	9	10	11	
	MeanNN	StdNN	RMSSD	NN50	pNN50	LF	HF	LFHFratio	SD1	SD2	ApEn	
1	888.1765	54.2176	29.5599		5	7.4627	57.8999	5.7270	10.1101	21.0778	73.9747	0.6664
2	627.8737	88.7645	49.5863		9	9.5745	53.4645	7.2263	7.3986	35.2521	121.0772	0.6404
3	850.8451	69.0761	36.4173		12	17.1429	73.0595	6.1211	11.9357	25.9431	94.4266	0.4912
4	475.3333	11.8632	6.0236		0	0	86.5944	6.8176	12.7016	4.2767	15.9806	0.6960
5	625.7917	22.4808	15.6021		1	1.0526	53.2089	31.4319	1.6928	11.0915	29.9405	0.6685
6	475.1587	19.6552	9.8460		1	0.8000	69.8838	6.3466	11.0113	6.9942	26.8845	0.5838
7	820.2192	57.6017	36.9222		15	20.8333	87.9518	7.7887	11.2922	26.3487	76.2682	0.3861
8	570.5524	34.9994	16.4042		1	0.9615	56.6435	5.3873	10.5142	11.6644	48.1296	0.7053
9	600.9400	80.9611	74.9213		35	35.3535	15.8946	16.7742	0.9476	53.2476	101.9840	0.6720
10	450.0602	24.9726	11.7844		1	0.7576	74.9029	11.4506	6.5414	8.3648	34.4390	0.7527
11	751.0750	75.5717	41.4532		19	24.0506	57.3054	3.9841	14.3834	29.4992	103.2417	0.5809
12	531.4336	71.4270	45.3937		15	13.3929	57.6818	26.9436	2.1408	32.2443	96.1386	0.5680
13	693.2558	52.4631	37.3586		7	8.2353	57.2500	14.8288	3.8607	26.5733	69.7389	0.5798
14	509.3729	17.7479	15.9657		3	2.5641	26.6040	17.4438	1.5251	11.3403	22.3736	0.5011
15	845.7183	68.1287	43.1082		17	24.2857	85.9519	11.4133	7.5309	30.7486	90.9511	0.3937
16	493.2066	31.1785	41.2630		15	12.5000	64.6236	28.2417	2.2882	29.3037	32.7647	0.8017
17	604.1818	32.3035	15.7978		1	1.0204	80.8418	12.6600	6.3856	11.2525	43.5930	0.5909
18	416.0694	27.8170	42.9685		17	11.8881	42.6699	49.7272	0.8581	30.4904	24.9236	0.6203
19	811.2432	45.6571	30.4870		9	12.3288	37.4006	12.6821	2.9491	21.7077	59.5919	0.5774
20	473.5433	29.1773	30.6080		14	11.1111	71.5783	20.4605	3.4984	21.7303	35.1431	0.5888
21	658.0220	35.0181	28.1504		4	4.4444	49.8081	31.2881	1.5919	20.0208	45.1749	0.6429
22	656.3261	43.9793	32.8981		12	13.1868	32.3114	28.3701	1.1389	23.4066	57.4134	0.6105

Figure 4.5: An example of input matrix. Observation are arranged in rows and HRV features in columns.

As output, the tool returns the HRV features that are good surrogates of the benchmark at the investigated time scales. The output is returned in the same format as the input matrices (Fig. 4.5).

In Chapter 5, this framework is applied to a specific case study: mental stress detection. The proposed framework is applied to identify ultra-short HRV features that are good surrogates of short term ones in detecting mental stress.

4.2.1.3 Matlab tool to investigate biomedical surrogates

In this section, a Matlab tool developed to support researchers in identifying the best surrogates based on the input data is presented (Fig. 4.6). The tool is targeted at researchers having familiarity with Matlab.

The UML sequence diagram of the tool is presented in Fig. 4.7 and 4.8.

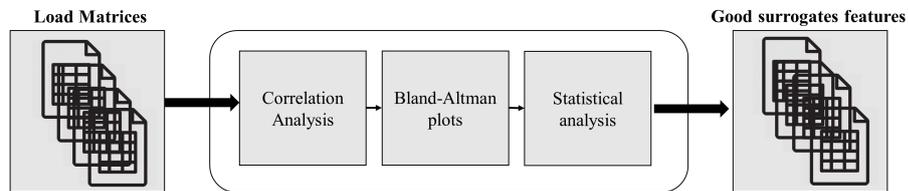


Figure 4.6: An overview of the developed tool to investigate surrogates. The inputs for the tool are matrices (i.e., HRV features extracted at different time scales) that the user wants to investigate. The tool consists of three main function blocks to run correlation analysis, investigate a Bland-Altman plot and explore statically significant differences across different time scales.

The Matlab code was checked for errors and standards compliance through debugging functions. It was also tested to refine the requirements until the design was fully functional and no unintended behaviours were encountered. The accuracy of each function (i.e., correlation functions, statistical analysis functions, etc.) was tested comparing the results with well-known and validated tools (e.g., IBM SPSS Statistics [232]).

To the best of the author's knowledge, this is the first tool developed in Matlab to investigate surrogates based on the frameworks presented in the above sections.

The output of the tool is given as general statistics of the input matrices and the final matrices containing only the features which are considered good surrogates for the benchmark (i.e., short term HRV features). The analysis typically takes from 2 to 5 minutes depending on the input size.

The main function is designed to receive user input whether the user wants to investigate surrogates for one condition (e.g., only resting) or two conditions (e.g., rest and stress conditions). Based on the user's response, the main function calls:

1. The “SurrogateOneCondition” function, which takes as input two (or more) matrices from .csv files (Fig. 4.5):
 - a benchmark matrix, with observations arranged in rows and features in columns and a header with features' names;
 - a matrix, which needs to be investigated (e.g., features extracted at different time scales), having the same size and structure as the benchmark matrix.

Then, it checks for normality (skewness) and if the features are non-normally distributed the user can decide whether to log-transform the features or proceed using non-parametric analysis. The function then calls:

- (a) The “Stat” function, which computes the mean, standard deviation and percentiles if normally distributed whereas it computes the median, standard deviation and percentiles if non-normally distributed.
- (b) The “Correlation” function, which performs parametric (i.e., Pearson's correlation) or non-parametric correlations (i.e., Spearman's correlation).
- (c) The “BlandAltmanPlts” function, which calls the function “BlandAltman” to produce BlandAltman plots using parametric or non-parametric analysis. The features that show a statistically significant correlation coefficient with the benchmark are then selected.

- (d) The “StatOneCondition” function, which performs parametric (i.e., Cohen's d function) or non-parametric (Cliff's delta function) analysis on the correlated features to investigate the effect size between the benchmark features and the ultra-short features.

At this stage, the features that show good agreement are selected as good surrogates of the benchmark.

2. The “SurrogateTwoConditions” function takes as input two (or more) matrices from .csv files:

- a benchmark matrix, with observations arranged in rows and features in columns, as the last column the class labels (in binary), and a header with features' names;
- a first matrix, which needs to be investigated (e.g., features extracted over different time scale, e.g., 3min) having the same size and structure as the benchmark matrix;
- a second matrix, which needs to be investigated (e.g., features extracted over different time scale, e.g., 1min) having the same size and structure as the benchmark matrix.

Then, it checks for normality (skewness) and if the features are non-normally distributed the user can decide whether to log-transform the features or proceed using a non-parametric analysis. The function then calls:

- (a) The “Stat” function, which computes the mean, standard deviation and percentiles and, performs t-test analysis if normally distributed whereas it computes the median, standard deviation and percentiles and performs a Wilcoxon Rank test analysis if non-normally distributed.
- (b) The “Correlation” function, which performs parametric (i.e., Pearson's correlation) or non parametric correlations (i.e., Spearman's correlation) between the benchmark and the ultra-short features for both rest and experimental conditions.
- (c) The “BlandAltmanPlts” function, which calls the function “BlandAltman” to produce BlandAltman plots using parametric or non-parametric analysis. Plots are produced for both conditions between the benchmark and the ultra-short features. The features that show a statistically significant correlation coefficient with the benchmark for both conditions are then selected.

- (d) The “Stat” function, which is then called again taking as input the correlated features and using parametric or non-parametric analysis to investigate which feature changes significantly between the two conditions for each time scale.
- (e) The “TrendAnalysis” function, which investigates if all the features maintain the same trends across different time scales.

At this stage, the features that maintain the same behaviour between the control and experimental conditions for the investigated time period windows are selected as good surrogates of the benchmark.

The Matlab scripts are reported in Appendix A, section A.2.

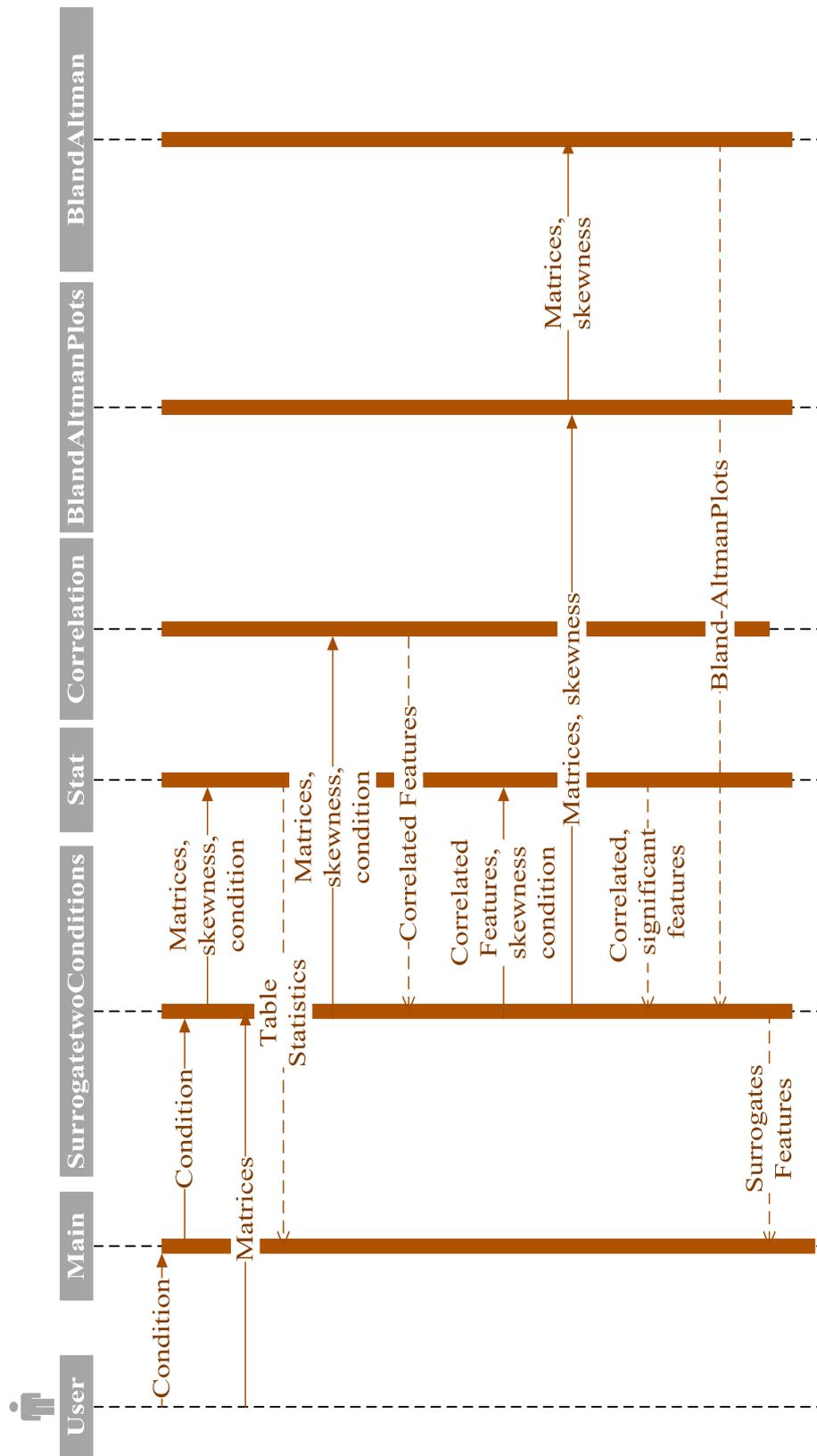


Figure 4.8: UML sequence graph of the Matlab tool for surrogate analysis in two conditions. The objects' blocks represent the functions, the straight arrows the inputs for each function and the dashed arrows the return outputs.

4.2.2 Data-driven machine learning techniques for biomedical signals in real-life settings

The diffusion of machine learning techniques in healthcare application areas is a subject of considerable ongoing research. Machine learning techniques provide methods and tools that can help solve diagnostic and prognostic problems across a variety of medical domains.

With the increase of healthcare services in real-life settings using vital signs provided by wearable sensors, the use of machine learning techniques is growing significantly. However, there are several unsolved challenges for the use of classification methods with small and unbalanced datasets. In fact, learning from a given data set to build a classification model becomes difficult when the available sample size is small or unbalanced. According to computational learning theory, the sample size in machine learning techniques has a major effect on the learning performance. Therefore, to build a correct classification model a sufficient amount of training data are required. However in the real world, there are many situations where the availability of data is restricted as some adverse events (e.g., falls, stroke, etc.) are rarely detected generating highly unbalanced datasets. In addition, the scarcity of good quality data in a real-life setting can generate small datasets. Furthermore, many events (e.g., stress, training performances, gait assessment) are investigated in-lab settings also generating small datasets which may lead to ineffective algorithms because of the lack of validation and testing procedures, that could give false or misleading information on the subject's status if applied to real-life settings.

Therefore, improved frameworks to refine already existing machine learning techniques for small and unbalanced datasets are presented in the next sections.

4.2.2.1 How to cope with small balanced datasets

A small dataset is very much a relative and subjective concept that needs to be defined. A dataset is considered small if it presents less than 10 occurrences per predictor variable [76]. This condition is characteristic of the biomedical engineering domain, where complexity and the high cost of experiments often constrain the number of available samples [233]. Another possible reason may be the miscalculation of the sample size during the study design. In fact, standard statistical methods mainly used to calculate the minimum sample size may be restrictive for modelling purposes. This problem is tackled in Chapter 5, section 5.3.1.1.

In the existing literature, different studies have tried to explore a way to develop accurate classifiers using a low sample size. Many of the methods proposed in the

literature manipulate the dataset to achieve high accuracy, but lose vital information and make communication with clinicians more complicated.

Existing methods in the literature One of the most used techniques to deal with small data is the use of artificial data to increase the sample size and increase the predicted accuracy in machine learning. In fact, adding some artificial data to the system in order to accelerate learning stability and to increase learning accuracy is one effective approach [234]. The virtual data concept is used in many small dataset learning methods. It was first proposed by Niyogi *et al.* [235] in the study of human face recognition. Another method based on multiple runs for model development and surrogate data analysis for model validation was proposed by Shaikhina *et al.* [236]. Surrogate data were generated from random numbers to mimic the distribution of the original dataset. This method seems to work for regression tasks based on small dataset. A more common technique to deal with small data is bootstrapping, which was first introduced by Efron [237]. Given a dataset of size n , a bootstrap sample is created by sampling n instances uniformly from the data (with replacement). However, bootstrapping can be shown to fail and bias the overall estimated accuracy in any required direction [238].

Other traditional methods to validate algorithms based on small datasets are the holdout approach, K-fold cross- and Leave-one-out (LOO) validations. The holdout approach partitions the data into two mutually exclusive subsets called training and test sets. It is common to design 2/3 of the data as training set and the remaining 1/3 as test set. Moreover, the holdout approach in random subsampling is repeated K times and the estimated accuracy is derived by averaging the runs. However, the main assumption of the independence of the instances in the test set from those in the training set is violated in random subsampling causing bias in the overall accuracy. Additionally, the holdout approach makes insufficient use of the data as usually a third of the data are not used for training the classifier. The other methods are the K-fold cross- and LOO validations. In the K-fold cross-validation approach, the dataset D is randomly split into K mutually exclusive subsets D_1, D_2, \dots, D_K of approximately equal size. The classifier then trains and tests K times; each time $t \in 1, 2, \dots, K$ is trained on the set $\frac{D}{D_t}$ and tested on the set D_t . Concerning LOO validation, the training is performed on the dataset of $n-1$ samples and tested on the remaining; it is repeated $n-1$ times. LOO is usually preferred for small datasets, however K-fold-cross-validation could also work by choosing an appropriate value for K , which depends on the number of samples available in the dataset.

Although some of these techniques have been shown to work on some datasets,

there are still some limitations to be addressed in order to avoid manipulation of the data, and generate general and reliable algorithms.

Proposed Framework Different from the existing methods, the proposed framework employs a set of rules to develop a reliable and general algorithm that does not lose vital information in the data and is easy for clinicians to understand.

The proposed framework consists of well-defined steps: splitting the datasets into separate folders, a feature selection process, then training, validating and testing the classifier.

Splitting of the dataset An important requirement in dealing with small datasets is an accurate feature selection based on the number of samples available for an independent dataset. In fact, the most common mistake in machine learning applications is to perform feature selection on the whole dataset. As a consequence, to reduce overfitting problems and bias in the overall accuracy of the classifier, the whole dataset can be randomly split per subject into two folders: Folder 1 (60 %) can be used for feature selection, for training and validating the classifiers; Folder 2 (40 %) for testing the classifier (Fig. 4.9). Although the best approach is to select the minimum set of features using a different folder from the one adopted to train the machine learning classifier [76], due to the small number of subjects, feature selection and training could be performed on the same folder.

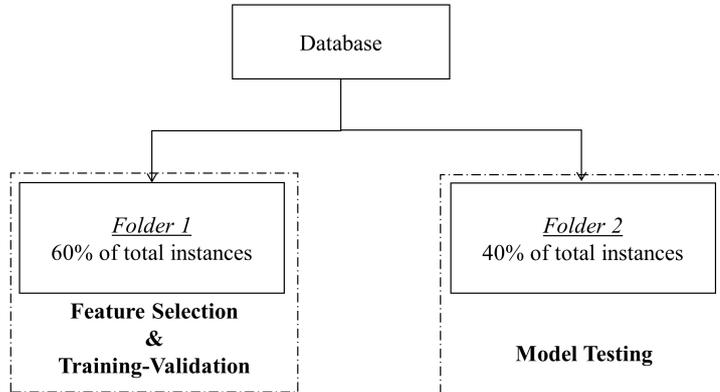


Figure 4.9: Splitting of the dataset into two folders. The whole dataset is split into two folders for feature selection, training, and testing respectively.

Feature Selection Feature selection is mainly used to limit the amount and dimensionality of the data or to select features that correlate well with the target class (observed healthcare event). Moreover, as a rule of thumb, for each predictor

variable at least 10 occurrences are necessary to avoid overfitting problems [76]. For instance, in order to cope with a binary problem of 40 occurrences per class, the maximum number of predictor variables should be 4. Therefore, to minimise the overfitting risk in a machine learning algorithm, the number of features used in the algorithm and its cardinality should be limited by the number of subjects presenting the event to detect.

In the existing literature, different feature selection methods are used and they can be subdivided into those that are unsupervised, i.e., unaware of class attributes (e.g., the removal of a feature with the same constant values throughout the whole dataset: PCA (Principal Component Analysis) and MF (Matrix Factorization)) and those that are supervised, i.e., driven by class information [239]. The latter group includes filter methods using information gain. However, the most common methods for feature selection are: the Wrapper, Relief Attribute Evaluation with Ranker and PCA. The Wrapper method takes into account class information by evaluating feature sets based on the performance of the classifier. Hence, the resulting feature set is tailored to a given classification method. The Wrapper method is the most ‘aggressive’ feature selection method. The Relief method is also supervised, but does not optimize feature sets directly for classifier performance. Thus, it takes into account class information in a ‘less aggressive’ manner than the Wrapper method. PCA is an unsupervised feature selection method and hence, it does not take into account class information at all and it is not easy for clinicians to interpret [240].

Therefore, a simple and robust approach to perform feature selection was proposed as shown in Fig. 4.10.

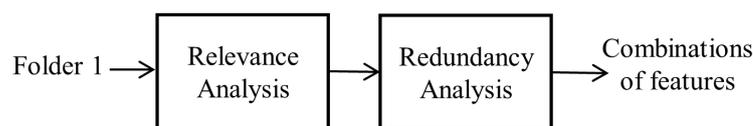


Figure 4.10: Framework for feature selection. The feature selection process is performed in a separate folder (Folder 1) and it consists of two main steps: relevance and redundancy analysis. Relevance analysis refers to a statistical significance test and redundancy analysis is explored via a correlation test.

As mentioned above, feature selection is performed in a separate folder (i.e., an independent part of the dataset) instead of using the whole dataset. In the case of small datasets, feature selection is performed in the same folder as the training, but in the case of bigger datasets, feature selection is performed in an independent

folder.

Feature selection is based on two main stages: relevance analysis and redundancy analysis. The relevance analysis refers to a statistical significance test such as a Wilcoxon Signed-Rank test in case of binary problems and non-normally distributed data and it aims to identify the features changing more significantly between two conditions. All those features changing significantly between the two conditions (p-value less than 0.05) are selected at this stage. All the relevant features (p-value<0.05) are then further minimised with the redundancy analysis aiming to exclude highly significant correlated features. Notions of measure redundancy are normally explored in terms of feature correlation. It is widely accepted that two features are redundant of each other if their values are strongly correlated. The features with a correlation coefficient (parametric or not parametric depending on the nature of the data) above a certain threshold (e.g., greater than 0.7) in absolute magnitude and with a significant p-value (less than 0.05) are considered highly correlated. In this final stage, only the combinations of features relevant and non-redundant are then considered for the next steps. Moreover, in the proposed framework the maximum number of features presented in each combination is selected according to the rule that for each predictor variable at least 10 occurrences are necessary to result in a classifier with reasonable predictive value as described in [76]. Therefore, in the case where a dataset contains 40 subjects presenting the event to be detected, the combinations of features could not contain more than 4 features.

The best combination of features is then selected as the one achieving the best performance during training. The significance and generality of this subset of features are finally validated in the remaining one or two folders, depending on the size of the available dataset.

It is important to note that Folder 1 before starting the feature selection, is an $M \times N$ matrix, with N equal to the number of features (typically N can range from 2 to 40) and M equal to the number of subjects (or observations) in this folder. After the feature selection, Folder' 1 contains all the combinations of relevant and non-redundant features ($M \times N'$ matrix, with N' less or equal to the number of features and M equal to the number of subjects in the folder).

Training, validation and testing Different machine learning methods can be used to develop classifiers aiming to automatically detect the event based on the selected combinations of features.

The training of the machine learning models and algorithm parameter tuning are performed in Folder' 1. Each of the methods should be used with all the

combinations of relevant and non-redundant features.

Moreover, Folder' 1 is also used to validate the models using a K-fold-cross-validation. The choice of the K-value is crucial in order to achieve high overall accuracy and reduce bias in the model. Therefore, as a rule of thumb, the K-value should allow each of the K-subsets to have at least 10 occurrences presenting the events to detect. For instance, if Folder' 1 contains 40 subjects presenting the events to be detected, the maximum number of the K-value should be equal to four. Stratified K-cross-validation (i.e., this means that each K-subset contains roughly the same proportion of the two types of class labels) is preferred to LOO, which is less robust as it induces fewer perturbations.

The models are then tested on an independent set (Folder 2) as shown in Fig. 4.11. Among the different machine learning methods used to train, validate and test, the best-performing model can be chosen as the classifier achieving the highest Area under the ROC Curve (AUC), which is a reliable estimator of both sensitivity and specificity rates. AUC was calculated as reported in Table 2.5. However, in the case where multiple classifiers have the same AUC, the model employing less number of features is chosen (i.e., less complex).

Binary classification performance measures are usually adopted according to the standard formulae reported in Chapter 2, section 2.4.2.

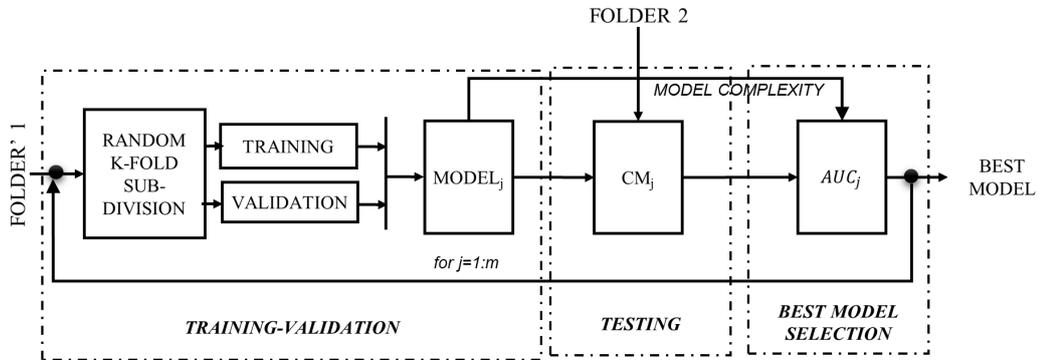


Figure 4.11: Model training, validation and testing for small balanced datasets. The training-validation procedure is repeated for each of the m machine learning methods used ($j=1, \dots, m$). For each machine learning methods used, the Confusion Matrix (CM_j) and the AUC_j are calculated. The best method is the one with the $\max(AUC_j)$.

The following procedure to develop an automatic classifier detecting an adverse event using a small dataset has been shown to be reliable (as presented in Chapter 5, section 5.2) and to not alter the data themselves as bootstrapping or the use of

artificial data may.

4.2.2.2 Matlab tool to develop an automatic classifier using small balanced datasets

In this section, a Matlab tool is developed to support researchers in performing feature selection, training, validating and testing using small datasets (Fig. 4.12).

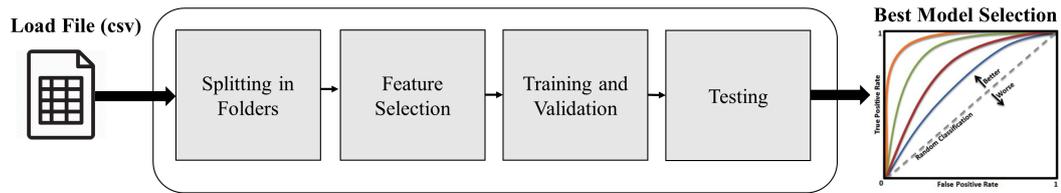


Figure 4.12: An overview of the developed tool to develop classifiers using small datasets. The tool consists of four main function blocks to split the dataset in input into two separate folders, run the feature selection process, train and validate the different machine learning methods considered and test the classifiers. The output of the tool is given as the best model to detect or predict an event.

The tool is targeted at researchers having familiarity with Matlab. The pseudocode for the tool is shown in Fig. 4.13.

```

Input file
Split dataset in two folders
Run feature selection process on Folder 1
    Run significance analysis
        if [normally distributed]
            then use t-test
        else [non-normally distributed]
            then use Wilcoxon rank test
        end
    Run correlation analysis on relevant features
        if [normally distributed]
            then use Pearson's correlation
        else [non-normally distributed]
            then use Spearman's correlation
        end
    Calculate the maximum number of features to use in the classification
    Generate all the possible combinations
For [each ML methods]
    For [all feature combinations]
        Train and validate on Folder' 1
    end
    Select the best features and classifier
    Test on Folder 2
end
Select the best methods based on AUC and classifier complexity

```

Figure 4.13: Pseudocode for the tool for small datasets. ML: Machine Learning, AUC: Area Under the Curve.

The UML sequence diagram for the complete tool is presented in Fig. 4.14. The feature selection process alone is presented in Fig. 4.15.

The Matlab code was checked for errors and standards compliance through debugging functions. It was also tested to refine the requirements until the design was fully functional and no unintended behaviours were encountered. The accuracy of each function (i.e., correlation analysis, statistical analysis, classification functions, etc.) was tested comparing the results with well-known and validated tools (i.e., IBM SPSS Statistics [232] and Weka for classifications [241]).

The main function is designed to call different sub-functions. The main sub-functions are responsible for splitting the datasets into folders, running feature selection, training, validating and testing small datasets. The outputs of the tool are given as the best classifier, confusion matrix, binary performance rates and ROC

curve.

In detail, the main function calls:

1. The “SplittingDataset” function, which takes as input a matrix (.csv file) with observations arranged in rows and features in columns. The file contains a header with features' names, the first column contains IDs of the subjects (i.e., anonymised number identification) and the last column the binary class labels. This function returns two folders split in 60% (‘Folder 1’) and 40% (‘Folder 2’) of the dataset for feature selection, training and testing respectively.
2. The “FeatureSelectionProcess” function, which performs feature selection on Folder 1 and gives in return all the possible combinations of relevant and non-redundant features (‘Feature Comb’). Based on the nature of the data (normally distributed or not), parametric or non-parametric analyses are performed accordingly. In fact, this function is mainly composed of:
 - (a) The “Stat” function, which performs parametric or non-parametric analysis to identify the features changing significantly between the two conditions (‘Relevant Features’).
 - (b) The “Correlation” function, which performs parametric or non-parametric correlation analysis and gives in return the correlation matrix with relevant and non-correlated features (‘Relevant, non-correlated Features’).
 - (c) The “MaxNumberofFeatures” function, which determines the maximum number of features for the specific dataset (‘MaxNumberFeatures’) (i.e.,
$$\frac{\text{number of subjects presenting the event to detect}}{10} \quad (4.1)$$
where 10 represents the number of occurrences necessary for each predictor variable).
 - (d) The “Redundancy” function, which takes as input the correlation matrix and based on the maximum number of features that can be used to develop the classifier, it generates all the possible combinations of relevant and non-redundant features (‘Feature Comb’).
3. The “GenerateTablestraining” function, which generates a structure of the matrices with all the best combinations of features (‘Folder 1: Training Data’).
4. The “TrainClassifier” function, which takes as input the matrices generated by “GenerateTablestraining” function and returns: the classifier (with validation), the best parameters for each classifier that minimise the estimated

cross-validation loss, the confusion matrices and ROC curves ('Feature Comb, AUC, Trained Classifier'). However, this function is called several times for the different machine learning methods used. This function calls:

- (a) The "MaxNumbCrossVal" function, which computes the maximum number of K-subsets that can be used during the validation process.
5. The "ClassifierSelection" function, which finds for each of the machine learning methods, the best classifier and the combination of HRV features ('Best Feature Comb') based on the AUC values and number of features employed by the classifiers (i.e., minimum complexity).
6. The "GenerateTablestesting" function, which generates matrices based on Folder 2 using the best combination of features ('Folder 2: Testing Data').
7. The "Testing" function, which tests the models in Folder 2. As output, the classifiers' performances ('Classifier Performance') and ROC curves are given.
8. The "ModelSelection" function, which selects the best model among the different machine learning methods used based on the AUC values and the number of features employed (i.e., minimum complexity).

This tool is highly adaptable to different biomedical problems and integration of other tools such as Weka [241] could be combined with the proposed tool to perform classification tasks.

The Matlab scripts are reported in Appendix A, section A.3.

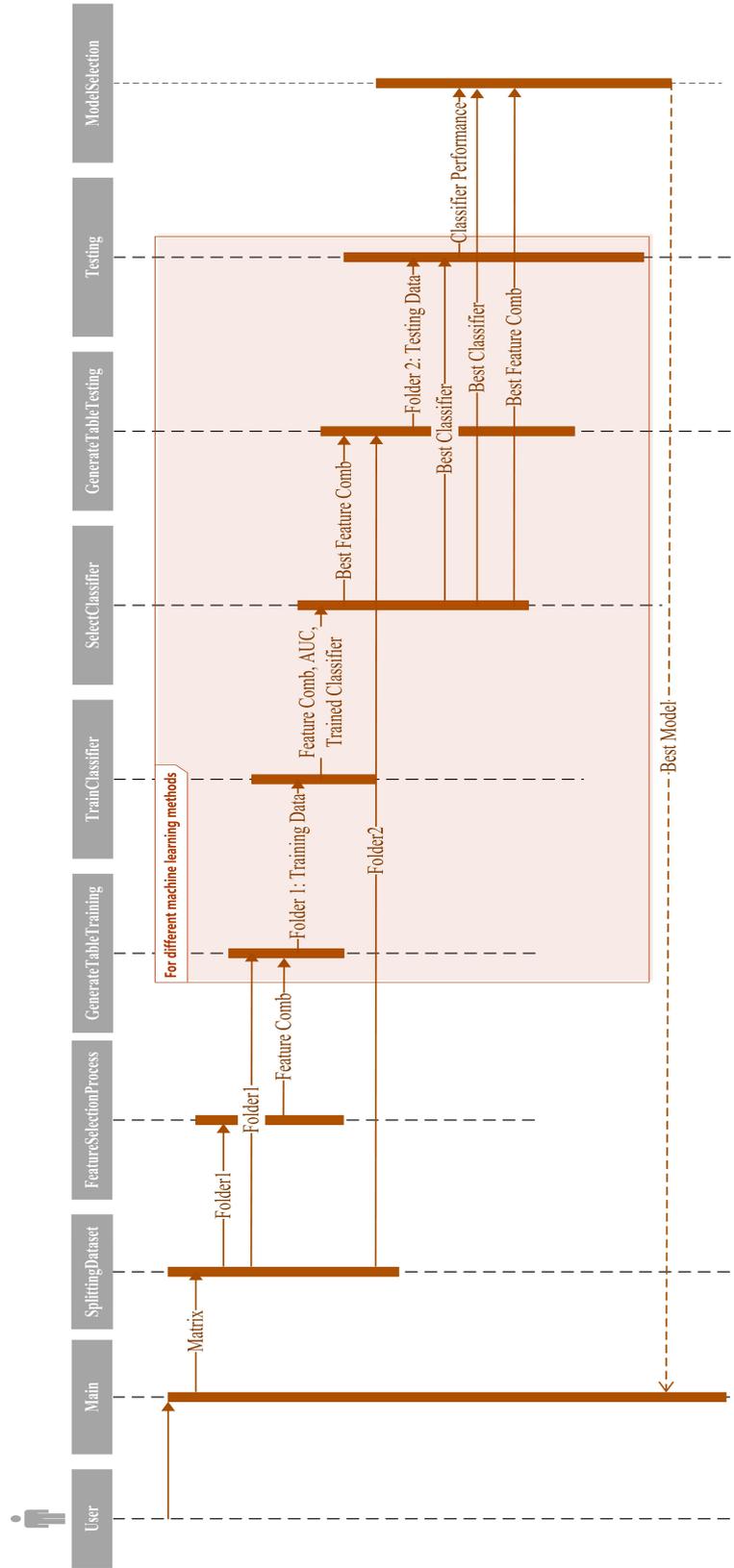


Figure 4.14: UML sequence graph of the Matlab tool for small datasets. The objects' blocks represent the functions, the straight arrows the inputs for each function and the dashed arrows the return outputs.

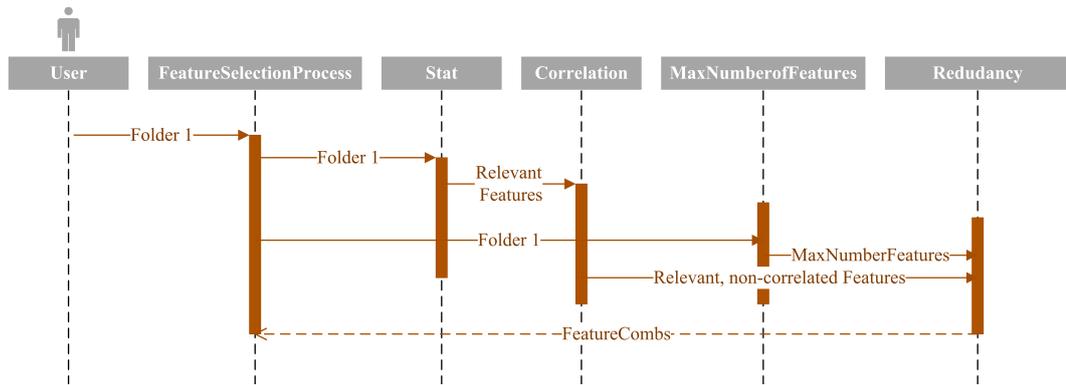


Figure 4.15: UML sequence graph of the Matlab tool for feature selection process. The objects' blocks represent the functions, the straight arrows the inputs for each function and the dashed arrows the return outputs.

4.2.2.3 How to cope with unbalanced datasets

A balanced dataset is very important in order to develop accurate and reliable classifiers. Typically real-world data are usually imbalanced and this is one of the main causes for the decrease of generalisation in machine learning algorithms. A dataset is said to be imbalanced if there are significantly more data points of one class and fewer occurrences of the other class [242]. One of the main reasons for unbalanced dataset in medicine is due to the difficulty of observing some events, also called rare events, such as falls, deaths and strokes. In fact, in medical datasets, high-risk patients tend to be the minority class. Most existing classification methods tend to not perform well on minority class examples when the dataset is extremely imbalanced. In fact, the sensitivity and specificity gap significantly reduces when the training set is balanced.

In the existing literature, different studies have tried to explore a way to develop accurate classifiers using unbalanced datasets. However, many of the methods proposed in the literature are more likely to overfit and less likely to be generalised.

Existing methods in the literature Sampling strategies have been used to overcome the class imbalance problem by either eliminating some data from the majority class (under-sampling) or by adding some artificially generated or duplicated data to the minority class (over-sampling), thereby manipulating the class distribution in the training set to maximise performance [243, 244]. In fact, these approaches incur the cost of overfitting or losing important information. Directed or focused sampling techniques select specific data points to replicate or remove. Japkowicz [245] proposed to resample minority class instances lying close to the class boundary, whereas

Kubat and Matwin [246] proposed resampling majority class such that borderline and noisy data points are eliminated from the selection. Yen and Lee [247] proposed cluster-based under-sampling approaches for selecting the representative data as training data to improve the classification accuracy. Liu *et al.* [248] developed two ensemble learning systems to overcome the deficiency of information loss introduced in the traditional random undersampling method. Chawla *et al.* [249] designed a sophisticated algorithm based on nearest neighbours to generate synthetic data for oversampling (SMOTE) and combined it with undersampling approaches and achieved significant improvements over random sampling techniques. Padmaja *et al.* [250] proposed an algorithm, called Majority Filter-based Minority Prediction (MFMP) achieving better performance than random resampling approaches. Estabrooks *et al.* [251] dealt with the rate of resampling required and proposed a combination scheme heavily biased towards under-represented class to mitigate the classifiers' bias towards the majority class.

At the algorithmic level, solutions include adjusting the costs of the various classes so as to counter the class imbalance, adjusting the probabilistic estimate at the tree leaf (when working with decision trees), adjusting the decision threshold, boosting, bagging and recognition-based (i.e., learning from one class) rather than discrimination-based (two class) learning [252]. However, these approaches can increase the likelihood of overfitting and discard potentially useful information which could be important for building rule classifiers.

Apart from these solutions, evaluation of the classifier for imbalanced datasets has always remained a big challenge. Provost and Fawcett [253] proposed the ROC convex hull method for estimating classifier performance whereas Kubat and Matwin [246] used the geometric mean to assess the classifier performance. However, the greatest disadvantage of these solutions is that they are very specific to the data and sometimes to the classification method.

Therefore, there are still many problems and challenges to be solved in order to develop reliable and accurate automatic classifier to detect rare events. A simple and effective framework to tackle the class imbalance problem for binary classifications in medicine is now proposed.

Proposed framework Different from existing methods, the proposed framework does not manipulate the data to achieve high performance, but it applies a set of rules reducing the risk of overfitting and dependency from the specific dataset used.

The proposed framework consists of well-defined steps: splitting the dataset into separate folders, a feature selection process, then training, validating and testing

the classifier.

In a different manner from the framework proposed to cope with small datasets, more attention is focused on the training and validation of unbalanced datasets.

Splitting of the dataset In the case where the dataset presents an adequate number of instances, the whole dataset can randomly be split per subject into three folders (Fig. 4.16): Folder 1 (usually the 34 %) can be used for feature selection; Folder 2 (39%) is used for training and validating the classification models; finally, Folder 3 (27 %) is adopted to evaluate the performance of the developed classification models. In the case of a highly imbalanced dataset, each folder should contain the same proportional percentage of minority instances.

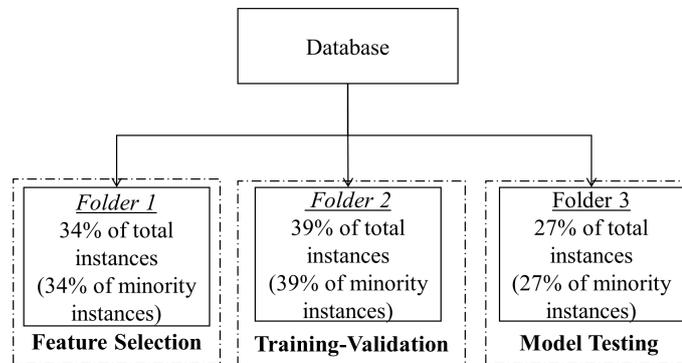


Figure 4.16: Splitting of the dataset into three folders. The whole dataset is split into three folders for feature selection, training and testing respectively. The percentage of instances (or subjects) should be the same for the majority and minority classes in each folder.

The subjects not included in Folder 1, are randomly assigned to Folder 2 or Folder 3 according to a K:2 ratio. The reason behind this splitting is that the remaining subjects are split into 2 folders according to the number of K-subsets used for cross validation [76]. K-value should be chosen carefully as it should allow for each K-subset to have at least 10 instances presenting the event to detect.

Feature Selection The number of features used in a machine learning algorithm should be strongly limited by the number of subjects presenting the event to detect in each folder, in order to minimise the risk of over-fitting. However, selecting the minimum set of features using the same folder utilised to train the machine learning algorithm can reduce the generalisability of the final decisional algorithm. Therefore, the features should be, where possible, minimised using only Folder 1.

The same procedure described in section 4.2.2.1 is also applied to this case. However, in the case of imbalanced datasets, the maximum number of features that can be used in the classification process is strongly limited to the number of subjects (i.e., belonging to the minority class) presenting the event to detect or predict.

Training, validation and testing Different machine learning methods can be used to develop classifiers aiming to automatically detect the event based on the selected combinations of features. Regarding algorithm parameters, they are tuned during training in Folder 2. Each of the methods should be used with all of the combinations of relevant and non-redundant features, coming from the analysis run using Folder 1. Given the relatively small and unbalanced number of events (minority class) in each K-subset, the K-cross-validation procedure is a critical step. In fact, although the cross-validation is stratified, the random allocation of one subject to one of the K-subsets can significantly alter the cross-validation estimates. Therefore, training and cross-validation procedures need to be repeated multiple times and tested for consistency. The cross-validation procedure needs to be repeated n times, with n equal to or greater than the number of instances belonging to the minority class, and cross-validation estimates averaged over those n iterations. This procedure needs to be performed for each machine learning method used to develop predictive algorithms, as shown in Fig. 4.17.

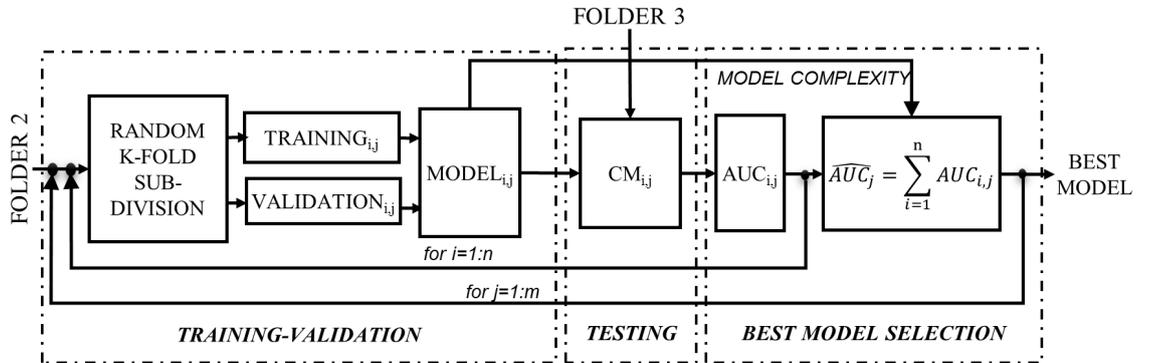


Figure 4.17: Model training, validation and testing for unbalanced datasets. The training-validation procedure is repeated n times ($i=1, \dots, n$) for each of the m machine learning methods used ($j=1, \dots, m$). For each iteration, the Confusion Matrix ($CM_{i,j}$) and the $AUC_{i,j}$ are calculated. The best method is the one with the max (\overline{AUC}_j).

Testing a classifier involves analysing its performances on a set of subjects that is independent of the training and validation set. Accordingly, Folder 3 is used

to test the trained algorithms. Finally, the best performing algorithm is selected as the one achieving the highest averaged AUC, which is a reliable estimator of both sensitivity and specificity rates and, in the case of an equal AUC average, the algorithm with minimal structural complexity (i.e., the minor number of employed features). To evaluate the classifier for imbalanced datasets, ROC curves and DOR can be employed.

In the case of a small unbalanced dataset, the same framework can be followed, splitting the dataset into two folders rather than three.

This methodology is applied to a specific case study: accidental falls prediction in later-life in Chapter 6.

4.2.2.4 Matlab tool to develop automatic classifier using unbalanced datasets

In this section, a Matlab tool is developed to support researchers in performing feature selection, training, validation and testing using unbalanced datasets (Fig. 4.18).

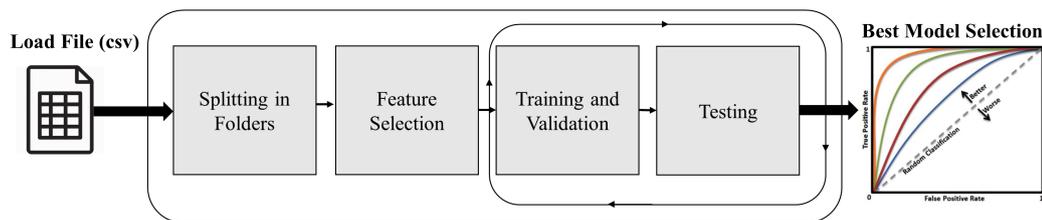


Figure 4.18: An overview of the developed tool to develop classifiers using unbalanced datasets. The tool consists of four main function blocks to split the dataset in input into three separate folders, run the feature selection process, train and validate different machine learning methods and test the classifiers. The training, validation and testing are iterated n times to reduce overfitting problems. The output of the tool is given as the best model to detect or predict an event.

The tool is targeted at researchers having familiarity with Matlab. The pseudo-code of the tool is shown in Fig. 4.19.

```

Input file
Split dataset in three folders
Run feature selection process on Folder 1
    Run significance analysis
        if [normally distributed]
            then use t-test
        else [non-normally distributed]
            then use Wilcoxon rank test
        end
    Run correlation analysis on relevant features
        if [normally distributed]
            then use Pearson's correlation
        else [non-normally distributed]
            then use Spearman's correlation
        end
    Calculate the maximum number of features to use in the classification
    Generate all the possible combinations
Calculate N: number of iterations
For [each ML methods]
    For [1:N]
        For [all feature combinations]
            Train and validate on Folder 2
        end
        Select the best features and classifier
        Test on Folder 3
    end
end
Select the best method based on AUC and classifier complexity

```

Figure 4.19: Pseudocode of the tool for unbalanced datasets. ML:Machine Learning, AUC: Area Under the Curve.

The UML sequence diagram of the complete tool is presented in Fig. 4.20.

The Matlab code was checked for errors and standards compliance through debugging functions. It was also tested to refine the requirements until the design was fully functional and no unintended behaviours were encountered. The accuracy of each function (i.e., correlation analysis, statistical analysis, classification functions, etc.) was tested comparing the results with well-known and validated tools (i.e., IBM SPSS Statistics [232] and Weka for classifications [241]).

The main function is designed to call different sub-functions. The main sub-functions are responsible for splitting the datasets into folders, running feature selection, training, validating and testing unbalanced datasets. The outputs of the

tool are given as the best classifier, confusion matrix, binary performance rates and ROC curve.

In detail, the main function calls:

1. The “SplittingDataset” function, which takes as input a matrix (.csv file) with observations arranged in rows and features in columns. The file contains a header with features' names, the first column IDs of the subjects and the last column the class labels (in binary). This function returns three folders split into 34% (‘Folder 1’), 39% (‘Folder 2’) and 27% (‘Folder 3’) for feature selection, training, and testing respectively. The same percentages are kept for the minority class in each folder.
2. The “FeatureSelectionProcess” function, which performs feature selection in Folder 1 and returns all the possible combinations of relevant and non-redundant features (‘Features Comb’). Based on the nature of the data (e.g., normally distributed or not), parametric or non-parametric analyses are performed accordingly. In fact, this function is mainly composed of:
 - (a) The “Stat” function, which performs parametric or non-parametric analysis to identify the features changing significantly between the two conditions.
 - (b) The “Correlation” function, which performs parametric or non-parametric correlation analyses and returns correlation matrix.
 - (c) The “MaxNumberofFeatures” function, which determines the maximum number of features for the specific dataset.
 - (d) The “Redundancy” function, which takes as input the correlation matrix and based on the maximum number of features that can be used to develop the classifier, it generates all of the possible combinations of relevant and non-redundant features.
3. The “GenerateTablestraining” function, which takes as input Folder 2 and generates a structure of matrices with all of the best combinations of features (‘Folder 2: Training Data’).
4. The “MinNumbN” function, which computes the minimum number of iterations (n), and based on the user input returns the number of iterations (‘N iteration’) that the training, validation and testing procedures need to be repeated.

5. The “TrainClassifier” function, which takes as input the matrices generated by “GenerateTablestraining” function and gives in return the classifier (with validation), the best parameters for each classifier that minimise the estimated cross-validation loss, the confusion matrices and ROC curves (‘FeatureComb, AUC, Trained Classifier’). This function calls:
 - (a) The “MaxNumbCrossVal” function, which computes the maximum number of K-subsets that can be used during the validation process.
6. The “ClassifierSelection” function, which finds for each of the machine learning methods the classifier (over n repetition) and the best combination of HRV features based on AUC values.
7. The “GenerateTablestesting” function, which generates matrices based on Folder 3 using the best combination of features selected previously (‘Folder 3: Testing Data’).
8. The “Testing” function, which tests the classifiers on Folder 3. As output, the classifiers' performances and ROC curves are given.
9. The “ModelSelection” function, which selects the best model among the different machine learning methods based on the averaged AUC values and number of features employed (i.e., minimum complexity).

This tool is highly adaptable to different biomedical problems and integration of other tools such as Weka [241] could be combined with the proposed tool to perform other classification tasks.

The Matlab scripts are reported in Appendix A, section A.4.

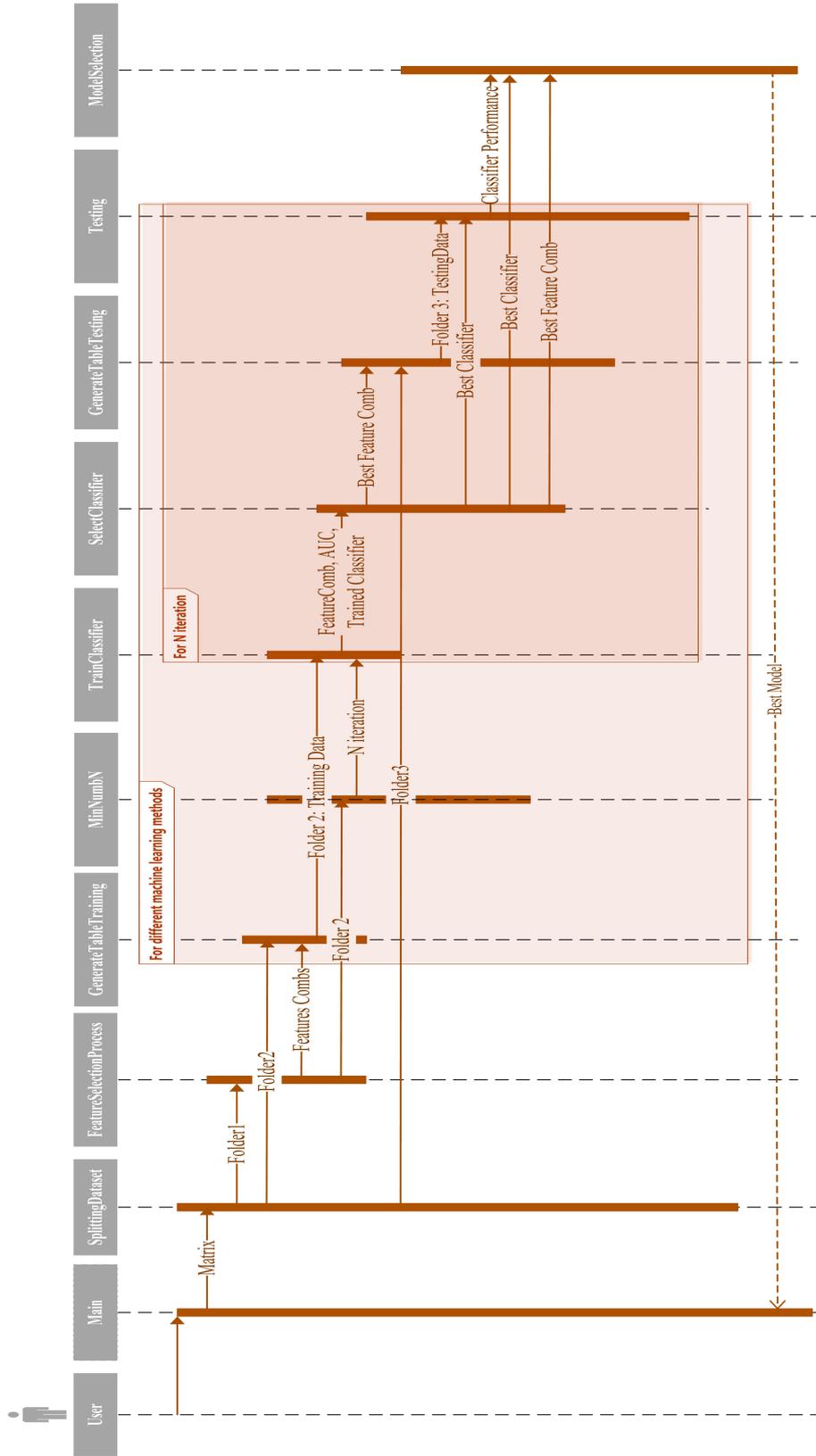


Figure 4.20: UML sequence graph of the Matlab tool for unbalanced datasets. The objects' blocks represent the functions, the straight arrows the inputs for each function and the dashed arrows the return outputs.

4.3 Conclusions and limitations

The first part of this chapter presented a novel approach to assess whether ultra-short HRV features are good surrogates of short term HRV features. As demonstrated in Chapter 3, section 3.2.3, there was an urgent need to develop a rigorous method to assess the validity and reliability of ultra-short HRV features. Although the proposed framework could be used for any application in which ultra-short and continuous monitoring of vital signs via wearable devices can be relevant, the presented framework is only adopted in regards to ultra-short HRV features and mental stress detection in Chapter 5.

The second part of this chapter presented improved frameworks to refine machine learning techniques for small and unbalanced datasets (sections 4.2.2.1 and 4.2.2.3). Although there are many algorithms coping with feature selection, small and unbalanced datasets, the presented frameworks attempt to overcome limitations that were identified in the existing algorithms. The proposed frameworks to develop automatic classifiers for small and unbalanced datasets have been mainly investigated in binary problems, but they could be extended to multiclass problems, even though they have not been tested yet.

Overall, the developed methods presented in this chapter have been applied to specific problems in Chapters 5 and 6 but they could also be used in other applications.

Chapter 5

Cardiovascular and Autonomic Response to Mental Stress

5.1 Chapter overview

The previous chapter presented the methodological frameworks and tools developed in an attempt to overcome the main limitations identified in the existing literature to monitor the ANS and the CVS in real-life settings.

In particular, in this chapter the framework to identify HRV surrogates in the ultra-short term and the approach to cope with small datasets are applied to a specific case study: the detection of mental stress via HRV. In fact, in this chapter, the relationship between the CVS and ANS is investigated as a means to detect acute mental stress.

Detection of mental stress via ultra-short HRV analysis is chosen as a case study, not only because is a burgeoning problem for modern society and causes alterations in both the CVS and the ANS, but also because the need of shortening HRV below 5 minutes (standard length) is a requirement for many off-the-self wearable sensors claiming to perform real-time stress detection. Nevertheless, shortening physiological signals - such as HRV - below standard recommendations may cause a loss of reliability and accuracy in the detection of mental stress.

The study workflow is presented in Fig. 5.1. A systematic review of the literature was carried out (in Chapter 3, section 3.2.2) to understand the relationship between HRV and ANS during stress, to extract significant HRV features and their pivot values and also to inform on future study designs (deliverable 1a). From the systematic review, it was clear that few studies investigated mental stress using ultra-short HRV features and even less assessed the validity of the latter before

investigating their discriminant values to detect stress. Therefore, a review of the existing methods to assess the validity of ultra-short HRV features was carried out (deliverable 1b) (please refer to Chapter 3, section 3.2.3). The results were surprising as none of the reviewed studies proposed a valid method to identify reliable subsets of ultra-short HRV features or surrogates of short term HRV features to allow the detection of the event of interest (e.g., stress). Therefore, in this chapter ultra-short HRV features are investigated during real-life stress and in-lab experiments (deliverables 1c and 1d) to identify ultra-short term HRV features that are “good” surrogates of short HRV features (i.e., ultra-short HRV features that maintain the same behaviour between the control and experimental conditions for the investigated time period windows) and to automatically detect mental stress in healthy subjects (deliverable 1f). Hence, four experiments were designed and carried out: stress assessment in real-life (E1); stress assessment in individual cognitive tasks (i.e., a Stroop Colour Word Test, E2); stress assessment in a group war scenario simulator (i.e., a war rescue mission in challenging virtual gaming, E3) and assessment of the impact of real and in-lab stressors (E4). From the experiments carried out in laboratory environments, the use of existing in-lab stressors turned out to be less stressful than real-life stress. Therefore, the power of in-lab data was explored (deliverable 1e) and a new model to detect mental stress in-lab was also developed to investigate the validity of ultra-short HRV features in a wider population.

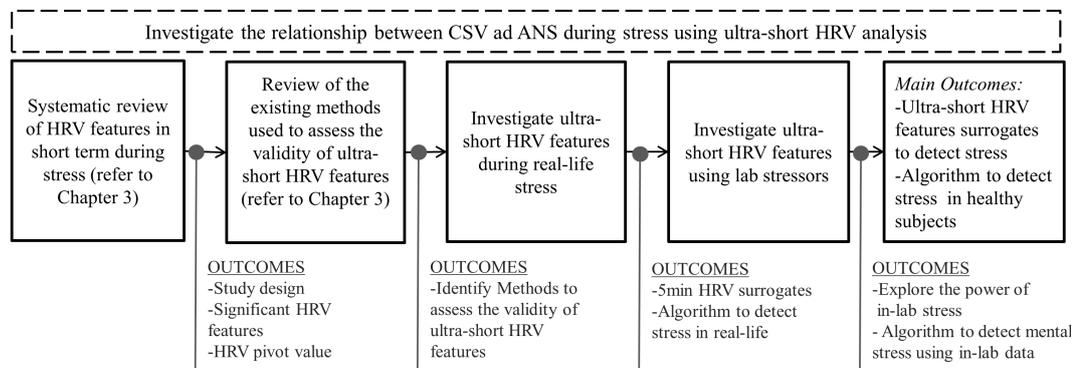


Figure 5.1: Workflow for Case Study 1. In order to investigate the relationship between the CVS and the ANS during mental stress, several steps have been undertaken to select ultra-short HRV features that are good surrogates of short term ones and develop an automatic classifier to detect stress via ultra-short HRV features.

In this chapter the main objectives 1 and 2, and the deliverables 1c, 1d, 1e are tackled.

5.2 Detection of mental stress in real life

Previous studies proved that long and short HRV features change consistently during mental stress and have shown to reliably capture mental stress in laboratory and real-life scenarios. However, much less work has been done on real-life stress detection via ultra-short term HRV analysis, but the demand for ultra-short term HRV analysis for monitoring an individual's well-being status is significantly increasing due to the proliferation of wearable sensors in the healthcare industry.

Therefore, the main goal of this study was to investigate which extent ultra-short HRV features are “good” surrogates of short HRV features (deliverable 1c) to automatically detect mental stress in real-life (deliverable 1f).

The data used for the first experiment (E1) were acquired prior to the work performed in this thesis and not by the author. This experiment was focused on data analysis and not on the generation of new data.

5.2.1 Dataset

The data were acquired from 42 students using a commercial electrocardiograph. The data acquisition was carried out in the School of Biomedical Engineering of the University Federico II, in Naples and therefore, approved by the Local Ethics Committee, as described in [45]. All the participants were healthy students at the University Federico II of Naples. The data were acquired on two different days within a month: the first recording was performed during an ongoing verbal AE (i.e., a stress session) before the Easter break (which in Italy lasts less than 10 days), while the second one was taken in a controlled resting condition (i.e., rest session) after the vacation. The resting condition was measured at the same hour of the day as for the stress session, in order to minimise circadian cycle effects on the HRV, and in the same menstrual cycle for women, as this is also a relevant measure for HRV features [254]. The participants were examined under standard conditions during rest and stress sessions: in the same quiet room, at a comfortable temperature, while sitting. Three lead ECG was recorded for at least 30 minutes for both the rest and stress sessions. The first 15 minutes (i.e., adaptation time) were excluded due to the high intensity of the real-life stressor and one ECG excerpt of 5 min was extracted and analysed. The investigators induced the participants to talk during the resting session, as they had done during the verbal examination, as talk has proven to alter respiration and therefore, HRV features. Participants were invited to refrain from drinking alcohol in the 24 hours before the data acquisition, and take no more than 2 cups of tea or coffee, as alcohol, caffeine and tea alter

HRV features. The participants enrolled had no history of heart disease, systemic hypertension, metabolic disorders or other diseases potentially influencing HRV. They were not obese and did not consume medication in the 24 hours preceding the experiments. All the participants signed specific informed consent form before the acquisition.

5.2.2 Hardware and software

Hardware The ECGs were acquired using a commercial electrocardiograph (Easy ECG Pocket, manufactured by Ates Medical), which allows 3 lead clinical research ECG acquisitions, with a sampling frequency of 500 Hz and a resolution of 12 bits per sample.

Software The different analyses were carried out using different software. The pre-processing of ECG signals was carried out using the PhysioNet's toolkit. Physio Toolkit is a large and growing library of software for: physiologic signal processing and analysis, detection of physiologically significant events using both classical techniques and novel methods based on statistical physics and nonlinear dynamics, interactive display and characterisation of signals, creation of new datasets, simulation of physiologic and other signals, quantitative evaluation and comparison of analysis methods, and analysis of nonequilibrium and nonstationary processes. Different functions were used for this study as detailed in section 5.2.4.

HRV analysis was carried out using the Kubios software. Kubios is an open-source software tool for studying the variability of heart beat intervals [32]. The final version of the software is compiled on a stand-alone C language application using the Matlab compiler Suite 2.3 and the free Borland C-Builder 5.5 compiler [32]. The graphic user interface (GUI) allows the user to analyse the data easily.

All the statistical analysis was carried out using in-house tools developed in Matlab2016b.

Machine learning algorithms were developed using the Weka Platform (version 3.6.10) and Matlab2016b software. Weka is issued by the University of Waikato as an open source software under the GNU General Public License [241]; it is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualisation. It is also well-suited for developing new machine learning schemes [241].

5.2.3 Data analysis

Fig. 5.2 describes the main stages of the data analysis carried out for this study. The ECGs were analysed and HRV features extracted from short excerpts (5 min, considered the benchmark in this study) and during shorter excerpts as detailed in section 5.2.4. The framework described in Chapter 4, section 4.2.1, was applied to the HRV features extracted from short and ultra-short time lengths in order to identify in a robust manner the ultra-short HRV features that are good surrogates of short term ones to detect stress. After having identified the subset of good surrogates, their discrimination power in detecting mental stress with sufficient accuracy was explored as detailed in section 5.2.7. The most common machine learning methods used in the existing literature on mental stress detection (please refer to Chapter 3, section 3.2) were used to develop an automatic classifier to detect stress via ultra-short HRV features. Support Vector Machine (SVM), Multilayer Perceptron (MLP), K-Nearest Neighbour (IBK), C4.5 and Linear Discriminant Analysis (LDA) were considered and developed using the Weka tool.

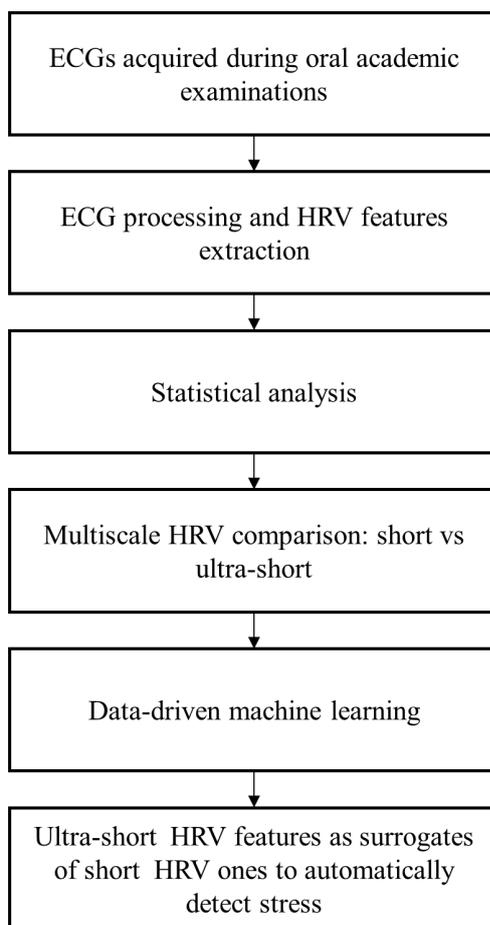
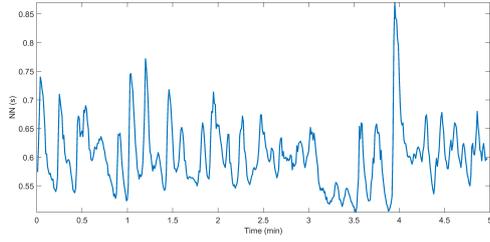


Figure 5.2: Data analysis flow for real-life stress. ECGs were acquired during a stressful situation, pre-processed and HRV features extracted. Statistical analysis identified HRV features that changed significantly during rest and stress conditions. Short and ultra-short HRV features were analysed to investigate the validity of ultra-short HRV analysis. Data-driven machine learning methods (i.e., Support Vector Machine (SVM), Multilayer Perceptron (MLP), K-Nearest Neighbour (IBK), C4.5 and Linear Discriminant Analysis (LDA)) were used to develop an automatic classifier to detect stress via ultra-short HRV features.

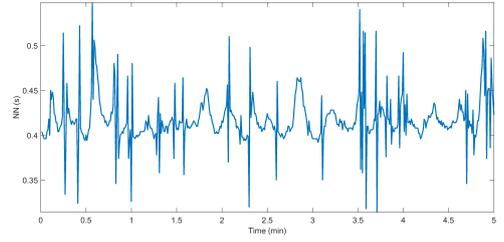
5.2.4 HRV analysis

As shown in Fig. 5.4, the RR interval time series were extracted from ECG records using an automatic QRS detector, WQRS, available in the PhysioNet's toolkit [255], based on nonlinearly scaled ECG curve length features [256]. An illustrative example of the raw NN (or RR, since no ectopic beats were detected) interval series over different time scales is shown in Fig. 5.3. However, no conclusions can be drawn

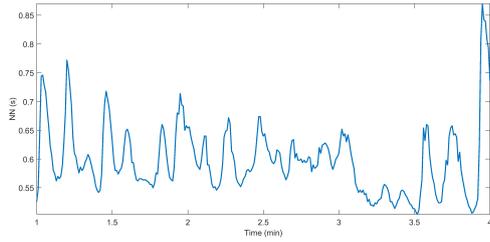
from these raw NN series, therefore, HRV features are then extracted and analysed.



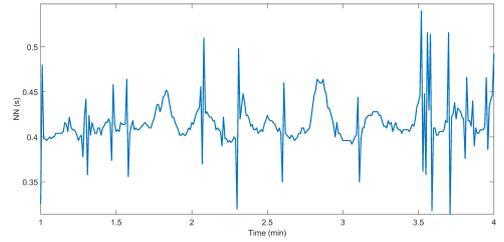
(a) Raw NN series at 5 min during rest session.



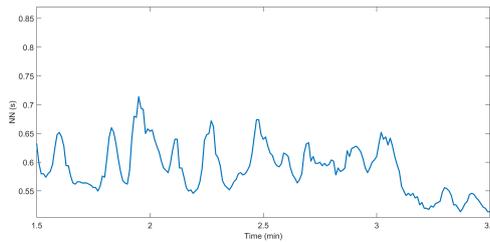
(b) Raw NN series at 5 min during stress session.



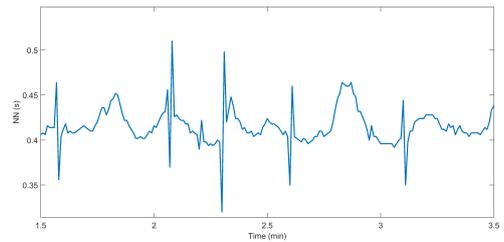
(c) Raw NN series at 3 min during rest session.



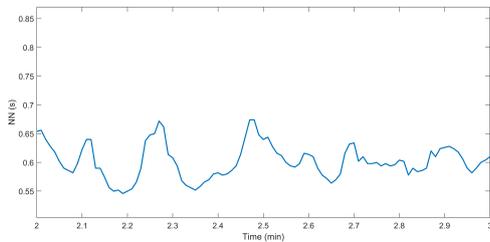
(d) Raw NN series at 3 min during stress session.



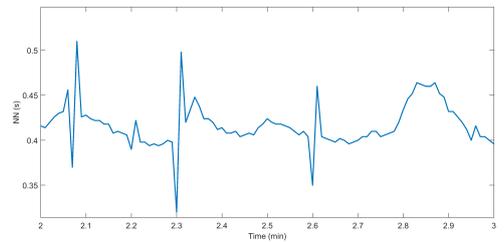
(e) Raw NN series at 2 min during rest session.



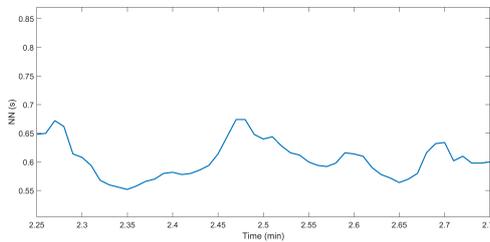
(f) Raw NN series at 2 min during stress session.



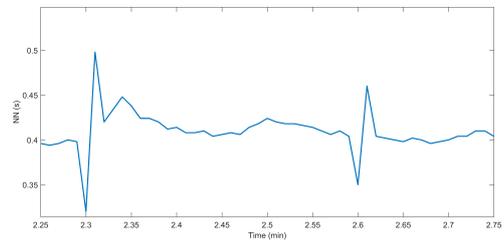
(g) Raw NN series at 1 min during rest session.



(h) Raw NN series at 1 min during stress session.



(i) Raw NN series at 30 sec during rest session.



(j) Raw NN series at 30 sec during stress session.

Figure 5.3: Raw NN series for one subject during the AE rest and stress sessions over different time scales (i.e., 5 min, 3 min, 2 min, 1 min and 30 sec.)

QRS review and correction were performed using WAVE, which is the graphical user interface to visualise a biomedical signal provided by PhysioNet and includes facilities for interactive annotation editing [257]. The automatic QRS detection was followed by manual review.

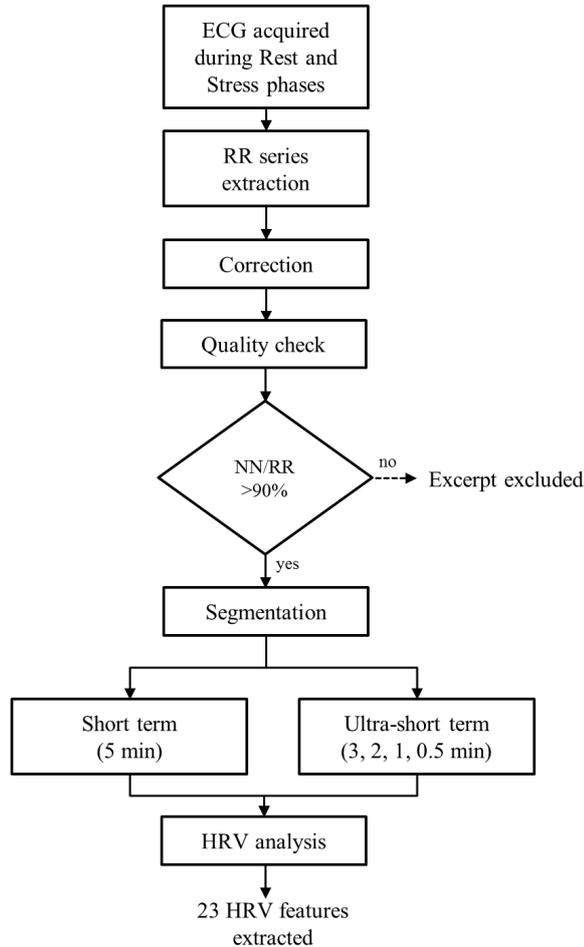


Figure 5.4: HRV processing workflow for real-life stress. NN/RR is the ratio of the total RR intervals labelled as NN (normal-to-normal beats). Short term HRV is analysed in 5 min excerpts. Ultra-short term HRV is analysed in excerpts of 3, 2, 1 and 0.5 min length.

The fraction of total RR intervals labelled as normal-to-normal (NN) intervals was computed as the NN/RR ratio. Since ectopic beat correction methods were not adopted and more than one RR excerpt was available for each subject, the NN/RR ratio was used to identify a window of time of sufficient quality, excluding those windows of time in which this ratio was lower than a threshold. Thresholds of 80% [255] and 90% [258] have been proposed. In the current study, in which subjects were

healthy and young, sitting in a comfortable position, a threshold of 90% was chosen and still no records were excluded. Therefore, short HRV features were computed from the first 5 min after the adaptation time for all of the subjects (Fig. 5.5).

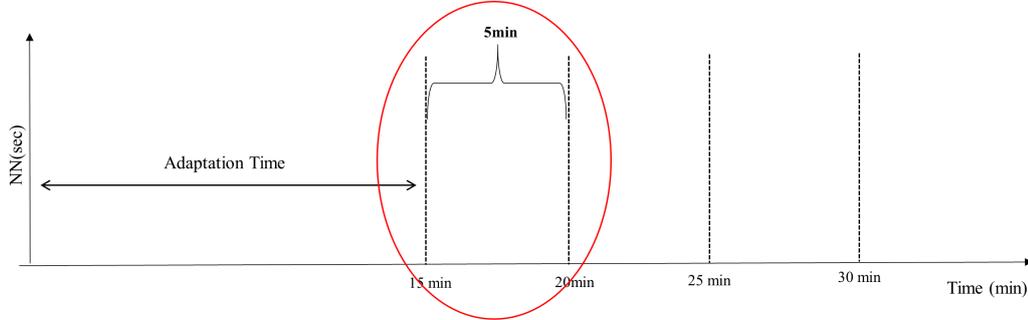


Figure 5.5: Excerpt extraction. 5min segments were extracted after the adaptation period for both rest and stress conditions.

The same 5 min excerpts were later used to extract shorter NN excerpts (Fig. 5.6, left-hand side) from which the ultra-short HRV features were computed. The initial choice of extracting the central excerpts was arbitrary. Therefore, in order to assess this choice, the shortest significant excerpt time length, resulting from the statistical significance and correlation analysis, was extracted from different locations within the 5 min excerpts (Fig. 5.6, right-hand side).

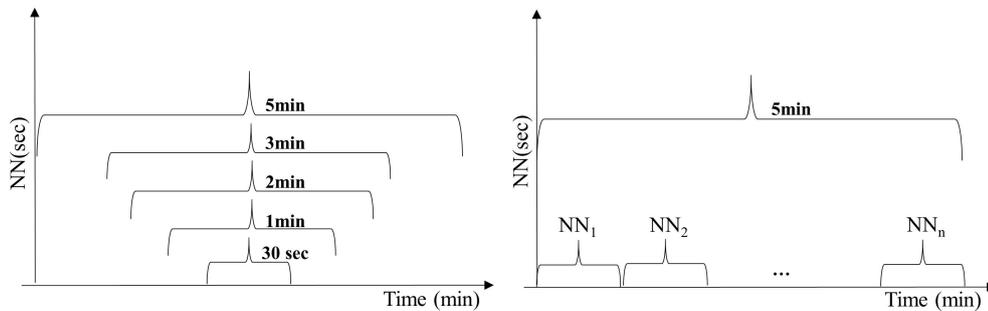


Figure 5.6: Segmentation process for real-life stress. The ultra-short HRV features were extracted from the central position of the 5 min NN excerpts (left-hand side). This procedure was repeated for the shortest significant length of NN excerpts. The shortest excerpt was extracted from different positions, without overlapping (right-hand side).

The HRV analysis was performed using the Kubios software [32]. Time, fre-

quency and non-linear features were analysed according to international guidelines (Chapter 2, section 2.4.1). In particular, frequency domain features were extracted from power spectra estimated with autoregressive (AR) model methods (Fig. 2.5). As reported in Table 5.1, 23 HRV features were extracted in the 5 min, 3 min, 2 min, 1 min, and 30 sec excerpts and subsequently analysed. However, not all the HRV features were computable in ultra-short time excerpts. In fact, it is generally recommended that spectral analyses are performed on recordings at least 10 times longer than the wavelength of the lower frequency limit that is at least 2 min for the Low Frequency power (LF). Therefore, LF was not computed for excerpts below 2 min along with LF/HF ratio and total power. Additionally, High Frequency power (HF) was not computed for excerpts below 1 min [15]. As far as non-linear HRV features are concerned, less has been explored in the existing literature. However, approximate entropy (ApEn) values were excluded for lengths below 3 min, since they have been shown to be unreliable [142]. Moreover, when the length of the data was reduced to 30 sec, most of the non-linear features became non-computable, due to the lack of samples.

Table 5.1: HRV features.

HRV Features	Units	Description
<u>Time Domain</u>		
MeanNN	[ms]	The mean of NN interval
StdNN	[ms]	Standard deviation of NN intervals
MeanHR	[1/min]	The mean heart rate
StdHR	1/min]	Standard deviation of instantaneous heart rate values
RMSSD	[ms]	Square root of the mean squared differences between successive NN intervals
NN50	-	Number of successive NN interval pairs that differ more than 50 ms
pNN50	[%]	NN50 divided by the total number of NN intervals
<u>Frequency Domain</u>		
LF	[ms ²]	Low Frequency power (0.04-0.15Hz)
HF	[ms ²]	High Frequency power (0.15-0.4 Hz)
LF/HF	-	Ratio between LF and HF band powers
TotPow	[ms ²]	Total power
<u>Non Linear Domain</u>		
SD1, SD2	[ms]	The standard deviation of the Poincare' plot perpendicular to (SD1) and along (SD2) the line-of-identity
ApEn	-	Approximate entropy
SampEn	-	Sample entropy
D2	-	Correlation dimension
dfa1, dfa2	-	Detrended fluctuation analysis: Short term and Long term fluctuation slope
RPlmean	[beats]	Recurrence plot analysis: Mean line length
RPlmax	[beats]	Recurrence plot analysis: Maximum line length
REC	[%]	Recurrence rate
RPadet	[%]	Recurrence plot analysis: Determinism
ShanEn	-	Shannon entropy

5.2.5 Statistical analysis

A normality test was run to show that the HRV features are non-normally distributed. Therefore, the median (MD), standard deviation (SD), 25th and 75th percentiles were calculated for all subjects to describe the distribution of the HRV features for the rest and stress sessions at 5 min, 3 min, 2 min, 1 min, and 30 sec.

5.2.6 Multi-scale HRV comparison: short VS ultra-short

The framework presented in Chapter 4, section 4.2.1.2 to identify ultra-short HRV features surrogates under two conditions (i.e., rest and stress conditions) was applied to this case study. Accordingly, a non-parametric statistical significance test and correlation analysis were performed in parallel as shown in Fig. 5.7, to select the subset of ultra-short HRV features that are good surrogates of short term HRV features.

5.2.6.1 Non-parametric statistical significance and trend analyses

The non-parametric Wilcoxon signed-rank test was used to investigate the statistical significances ($p\text{-value}<0.05$) of the HRV features' variation between the stress and rest sessions for each excerpt length (i.e., 5 min, 3 min, 2 min, 1 min, and 30 sec). The trends of the HRV features were also reported, where possible, using the following convention:

- two arrows, \Downarrow (or \Uparrow), were used to report a significant ($p\text{-value}<0.05$) decrease (or increase) of a feature during the stress session;
- one arrow was used for non-significant variations: \downarrow (or \uparrow) indicated a non-significant ($p\text{-value}>0.05$) decrease (or increase) of a feature during the stress session.

Trend analysis consisted of inspecting any change in the trends of the HRV features across time scales. An HRV feature was assumed to maintain the same behaviour across the 5 different time-scales (5 min, 3 min, 2 min, 1 min, and 30 sec) if:

1. the Wilcoxon's test p -value was less than 0.05 between rest and stress conditions at each time-scale;
2. the ultra-short HRV feature's trend changed between the rest and stress sessions consistently with the equivalent short HRV feature's trend.

5.2.6.2 Correlation analysis and Bland-Altman plots

The Spearman's rank correlation was used to investigate to what extent an ultra-short HRV feature was correlated with the equivalent short HRV feature. Spearman's rank correlation was used as HRV features are non-normally distributed. In fact, Spearman's rank correlation is a non-parametric test used to measure the statistical dependence between the rankings of two variables and does not make any assumptions about their distribution. Spearman's rank correlation is described by Spearman's correlation coefficient (ρ) measuring the strength and direction of the association between two ranked variables. The threshold limit for ρ was set to 0.7; ρ greater than 0.7 describes a strong association between two ranked variables [227]. The statistical significance of this association is demonstrated by a p -value (p_ρ) lower than 0.05.

As a first step, each ultra-short HRV feature was investigated against the equivalent short HRV feature during a resting condition (Fig. 5.7). Secondly, each ultra-short HRV feature was also explored during a stress condition.

However, a correlation coefficient is blind to the possibility of bias caused by the difference in the mean or median between two measurements, more specifically a strong correlation does not necessarily imply a close agreement. Therefore, Bland-Altman procedure was used to calculate the 95% Line of Agreement (LoA) [130]. Although, the Bland-Altman procedure works better with normalized data, non-parametric methods were used to assess the degree of agreement between short and ultra-short HRV features. In fact, median and Interquartile Range (IQR) were calculated and Wilcoxon's rank test was used to estimate the p-value. In contrast to the traditional Bland-Altman plots, the measurements of the 5 min excerpts were plotted on the x-axis [156]. The bias was calculated as the median difference between the HRV features at 5 min and the ultra-short HRV features.

5.2.6.3 Feature subset selection

At this stage, it was assumed that an ultra-short HRV feature was a good surrogate of the equivalent short term one, only if:

- the feature maintained the same behaviour between rest and stress at each time scale as detailed above;
- the ultra-short HRV feature was highly and significantly correlated (i.e. $\rho > 0.7$ and $p_\rho < 0.05$) with the corresponding short HRV term feature, across all the time scales in both the rest and stress sessions.

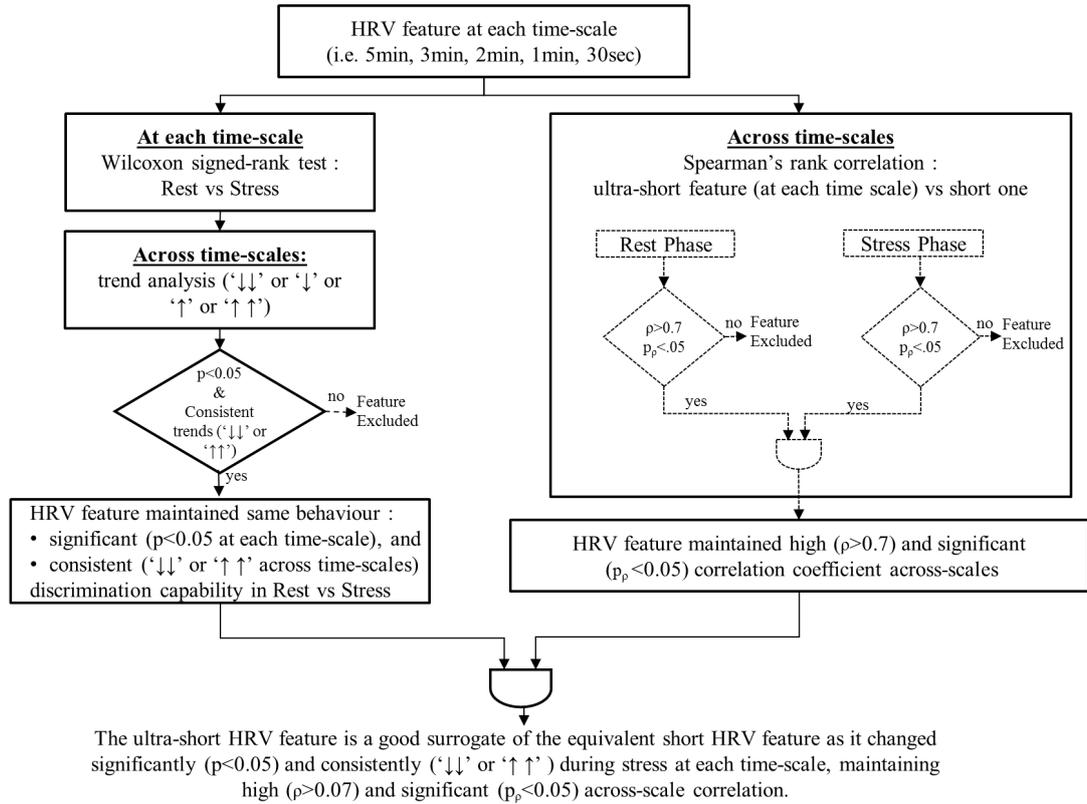


Figure 5.7: Methodological workflow for the identification of the good surrogates. HRV features at each time scale were analysed via a statistical significance test (p : p -value < 0.05). HRV features were also investigated with trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) under stress ($p > 0.05$). Only the features that maintained the same behaviour were selected. Moreover, HRV features were also investigated via correlation analysis across time scales in both the rest and stress conditions. HRV features that showed to be significantly and highly correlated were selected (ρ : Spearman's rank coefficient > 0.7 ; p_p : Spearman's rank p -value < 0.05). The HRV features that maintained the same behaviour and highly correlated across time scales were selected as good surrogates.

5.2.7 Data-driven machine learning

Short HRV features (benchmark) were used to train, validate and test an automatic classifier to detect mental stress. The performance of this classifier was then tested inputting ultra-short HRV features to assess the discriminant power of ultra-short term HRV analysis.

The framework introduced in Chapter 4, section 4.2.2.1 to cope with small dataset was applied to this study. Accordingly, the whole dataset was split per

subject into two folders: Folder 1 (60%) was used for feature selection, for training and validating the classifiers; Folder 2 (40%) was used to test the model.

5.2.7.1 HRV feature selection

In order to minimise the over-fitting risk in the machine learning models, the number of features used in the model and its cardinality was limited by the number of subjects presenting the event to detect (i.e., stress) [76].

Although the best approach is to select the minimum set of features using a different folder from the one adopted to train the machine-learning model, due to the small number of subjects in this study, feature selection and model training were performed on the same folder (Folder 1: 25 subjects).

Because 5 min is defined as the standard length for short term HRV analysis, the feature selection process was performed using 5 min HRV excerpts accordingly applied to the proposed framework presented in Chapter 4, section 4.2.2.1. Among the 23 HRV features initially computed, only those that showed to be good surrogates in the ultra-short time excerpts entered the feature selection process. Therefore, the dataset contained in Folder 1 before starting the features selection, was a $N \times M$ matrix, with N equal to the number of good surrogate features and thus less than or equal to the number of HRV features initially extracted (i.e., $N \leq 23$) and M equal to the number of subjects in this folder (i.e., 25). The feature selection was based on two main stages: the relevance analysis and the redundancy analysis. The former was performed using a non-parametric statistical test, which aimed to identify the HRV features changing more significantly across the stress and rest sessions. Since not all the HRV features were normally distributed, (i.e., frequency features have non-symmetric distributions) the Wilcoxon signed-rank test was adopted. All the HRV features changing significantly between the stress and rest sessions (p-value less than 0.05) were selected at this stage. All the relevant HRV features (p-value < 0.05) were then further minimised with the redundancy analysis aiming to exclude highly correlated features. Therefore, only one feature from each cluster of features mutually correlated was selected using Spearman's rank correlation. After the feature selection, Folder' 1 presented all the possible combinations of relevant and non-redundant HRV features.

5.2.7.2 Machine learning methods

The five most commonly applied machine-learning methods were used to develop a classifier aiming to automatically detect mental stress based on short term HRV

features. The machine learning methods were: a Support Vector Machine (SVM), which belongs to a general field of kernel-based machine learning methods and is used to efficiently classify both linearly and non-linearly separable data; a Multilayer Perceptron (MLP), which consists of an artificial neural network of nodes (processing elements) arranged in layers ; a K-Nearest Neighbour (IBK) approach, which finds a group of K objects in the training set that are closest to the test object, bases the assignment of a label on the predominance of a particular class in the neighbourhood; C4.5, which builds decision trees from a set of training data, using the concept of information entropy; Linear Discriminant Analysis (LDA), which aims to find linear combinations of the input features that can provide an adequate separation between two classes.

Regarding the model parameters, for the MLP classifier, the learning rate (LR) ranged from 0.3 to 0.9, the momentum (M) from 0.2 to 1 and the number of excerpts (NE) from 100 to 2000 [259]; for the SVM, polynomial kernel function was used, varying the degree (E) from 1 to 5 [260, 261]; for IBK, K was varied from 1 to 5 [262]. C4.5 trees were developed by varying confidence factor (CF) for pruning from 0.05 to 0.5 and the minimum number of instances per leaf (ML) from 2 to 20 [263]. The model parameters were tuned during training in Folder' 1. The best parameters for each method were chosen as the ones that optimise their overall accuracy.

Each of these methods was used with all the combinations of relevant and non-redundant HRV features. The possible combinations counted N out of the selected features, with N spanning from 3 to 4 features, as the subjects presented the event to be detected in the dataset were 42.

5.2.7.3 Training, validation and testing

The training of the machine-learning methods (including feature selection and model parameter tuning) was performed on the Folder' 1 (25 subjects) and using 5 min HRV features (benchmark). Folder' 1 was further divided into 3 equal sized sub-samples, according to the 3-fold person-independent cross-validation approach.

The model was then tested on Folder 2 (17 subjects) using the short and ultra-short HRV features to assess their efficacy in automatically detecting mental stress.

Binary classification performance measures were adopted according to the standard formulae reported in Chapter 2 , section 2.4.2. Among the five different machine-learning methods used to train, validate and test the classifiers (SVM, MLP, IBK, C4.5, and LDA), the best-performing model was chosen as the classifier achieving the highest Area under the Curve (AUC), which is a reliable estimator of both sensitivity and specificity rates. In addition, the ROC curve for the best model was

constructed.

5.2.8 Results

ECGs recorded from 42 healthy subjects were analysed in the current study. Subjects with age from 18 to 25 years old were no obese (Body Mass Index (BMI) 22.3 ± 2.7) and they were not taking any medication for the duration of the study.

5.2.8.1 Statistical analysis

HRV features median (MD), standard deviation (SD), 25th and 75th percentiles and p-value were calculated on 5 min, 3 min, 2 min, 1 min, and 30 sec NN data series and presented in Tables 5.2, 5.3, 5.4, 5.5, 5.6 respectively.

Table 5.2: HRV features in rest and stress from 5 min NN data series. AE experiment.

Short term: 5 min										
	Rest				Stress					
HRV features	MD	SD	25 th	75 th	MD	SD	25 th	75 th	p-value	Trend
MeanNN (ms)	726.988	83.695	647.623	776.4	483.594	66.464	446.02	512.257	0.000	↓↓
StdNN (ms)	59.642	18.383	44.455	70.00	36.619	16.266	26.438	46.990	0.000	↓↓
MeanHR (1/min)	82.976	9.704	78.038	93.13	125.242	16.711	117.42	135.807	0.000	↑↑
StdHR (1/min)	6.378	1.812	5.579	7.234	9.084	3.350	6.997	11.509	0.000	↑↑
RMSSD(ms)	33.806	14.719	22.848	42.49	34.120	16.643	18.069	46.997	0.996	↑
NN50 (-)	49.500	41.888	17.000	80.00	55.000	68.568	12.000	102.000	0.499	↑
pNN50 (%)	12.332	11.336	3.899	19.72	9.218	10.004	1.984	17.354	0.341	↓
LF (ms ²)	1661.164	1252.05	995.772	2497.	454.370	874.45	186.01	1070.18	0.000	↓↓
HF (ms ²)	381.739	488.299	212.089	591.2	141.648	233.87	67.719	344.740	0.000	↓↓
LF/HF (-)	4.646	2.512	2.811	6.424	3.267	2.181	1.943	4.662	0.011	↓↓
TotPow (ms ²)	3313.660	2160.75	1953.02	4694.	1045.95	1709.4	427.80	2214.64	0.000	↓↓
SD1 (ms)	23.936	10.423	16.172	30.08	24.146	11.777	12.788	33.267	0.996	↑
SD2 (ms)	79.555	24.420	61.047	94.59	46.969	22.878	31.713	57.903	0.000	↓↓
ApEn (-)	1.103	0.125	1.020	1.193	0.933	0.240	0.842	1.178	0.012	↓↓
SampEn (-)	1.325	0.272	1.111	1.546	0.876	0.392	0.760	1.228	0.000	↓↓
D2 (-)	3.179	1.090	2.245	3.544	1.496	1.283	0.469	2.575	0.000	↓↓
dfa1 (-)	1.439	0.161	1.283	1.511	1.043	0.446	0.690	1.447	0.000	↓↓
dfa2 (-)	0.716	0.183	0.644	0.954	0.767	0.136	0.679	0.852	0.862	↑
RPImean (beats)	10.439	2.479	9.519	12.68	13.326	6.771	11.105	16.920	0.002	↑↑
RPImax (beats)	282.000	111.223	178.000	384.0	179.000	136.59	86.000	282.000	0.004	↓↓
REC (%)	32.570	6.276	29.553	37.59	43.252	12.050	36.107	49.023	0.000	↑↑
RPadet (%)	98.776	0.858	98.314	99.20	99.254	1.277	98.138	99.633	0.034	↑↑
ShanEn (-)	3.139	0.233	3.044	3.363	3.418	0.398	3.210	3.642	0.001	↑↑

MD.: Median; SD: Standard Deviation; trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) under stress ($p > 0.05$). In bold HRV features changing significantly between rest and stress conditions.

Table 5.3: HRV features in rest and stress from 3 min NN data series. AE experiment.

Ultra-short term: 3 min										
HRV features	Rest				Stress				p-value	Trend
	MD	SD	25 th	75 th	MD	SD	25 th	75 th		
MeanNN (ms)	725.245	82.975	651.83	773.57	482.705	67.077	443.36	522.57	0.000	↓↓
StdNN (ms)	57.709	17.957	41.214	69.298	35.211	17.773	26.797	46.104	0.000	↓↓
MeanHR (1/min)	83.367	9.644	78.045	92.336	124.860	17.098	117.070	136.07	0.000	↑↑
StdHR (1/min)	6.251	1.866	5.397	7.449	8.943	3.488	6.452	12.57	0.000	↑↑
RMSSD (ms)	33.283	14.465	24.099	42.016	34.344	16.637	18.856	46.81	0.764	↑
NN50 (-)	28.500	24.984	10.000	44.000	35.500	40.741	7.000	66	0.359	↑
pNN50 (%)	11.941	11.087	3.831	19.731	9.968	9.954	1.728	17.5	0.418	↓
LF (ms ²)	1397.950	1475.536	569.630	2224.400	415.345	852.459	106.490	889.28	0.000	↓↓
HF (ms ²)	297.630	442.760	173.440	516.530	101.281	281.341	56.541	291.13	0.000	↓↓
LF/HF (-)	4.597	2.746	2.912	5.674	3.607	3.353	1.553	5.126	0.048	↓↓
TotPow (ms ²)	2529.000	2154.584	1845.000	4745.000	735.500	2056.264	286.000	1642	0.000	↓↓
SD1 (ms)	23.583	10.253	17.081	29.774	24.317	11.779	13.349	33.147	0.785	↑
SD2 (ms)	77.528	23.960	54.410	90.998	40.775	25.432	30.339	58.666	0.000	↓↓
ApEn (-)	0.987	0.093	0.923	1.039	0.870	0.192	0.780	1.049	0.041	↓↓
SampEn (-)	1.350	0.258	1.173	1.531	0.928	0.420	0.701	1.275	0.000	↓↓
D2 (-)	3.045	1.069	2.044	3.385	1.398	1.208	0.331	2.417	0.000	↓↓
dfa1 (-)	1.440	0.190	1.302	1.543	1.075	0.468	0.653	1.382	0.000	↓↓
dfa2 (-)	0.728	0.188	0.614	0.875	0.761	0.170	0.624	0.867	0.911	↑
RPlmean (beats)	10.171	2.876	8.898	13.569	13.224	7.291	10.427	16.665	0.023	↑↑
RPlmax (beats)	179.000	65.203	110.000	230.000	141.000	96.181	74.000	201	0.117	↓↓
REC (%)	33.058	7.586	27.676	38.662	42.708	14.129	32.762	50.409	0.003	↑↑
RPadet (%)	98.774	0.935	98.097	99.249	99.195	1.541	97.826	99.632	0.074	↑
ShanEn (-)	3.089	0.264	2.943	3.332	3.361	0.428	3.131	3.5907	0.009	↑↑

MD: Median; SD: Standard Deviation; trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) under stress ($p > 0.05$). In bold HRV features changing significantly between rest and stress conditions.

Table 5.4: HRV features in rest and stress from 2 min NN data series. AE experiment.

HRV features	Ultra-short term: 2 min									
	Rest					Stress				
	MD	SD	25 th	75 th	MD	SD	25 th	75 th	p-value	Trend
MeanNN (ms)	720.278	97.448	651.239	767.654	477.222	69.265	440.625	528.335	0.000	↓↓
StdNN (ms)	49.102	19.016	39.252	69.378	34.953	18.357	26.144	43.930	0.000	↓↓
MeanHR (1/min)	83.786	14.068	78.615	92.636	126.107	17.810	115.474	137.403	0.000	↑↑
StdHR (1/min)	6.147	2.044	5.174	7.253	8.481	3.560	6.715	12.470	0.000	↑↑
RMSSD (ms)	32.819	14.947	20.201	39.421	34.791	17.349	17.536	47.076	0.986	↑
NN50 (-)	16.000	16.992	7.000	26.000	23.000	28.112	3.000	42.000	0.496	↑
pNN50 (%)	9.650	11.153	3.784	16.129	10.333	10.147	1.038	16.342	0.452	↑
LF (ms ²)	1359.861	1362.08	736.544	2559.88	416.867	997.892	145.482	836.823	0.000	↓↓
HF (ms ²)	312.068	443.887	165.133	564.119	134.815	280.732	61.680	264.583	0.000	↓↓
LF/HF (-)	4.498	2.977	2.997	5.899	3.321	3.195	1.517	5.093	0.031	↓↓
TotPow (ms ²)	2460.120	2046.27	1637.19	4269.09	814.113	2180.51	305.468	1735.69	0.000	↓↓
SD1 (ms)	23.275	10.610	14.323	27.963	24.651	12.291	12.430	33.350	1.000	↑
SD2 (ms)	66.157	25.363	53.775	93.058	38.650	26.374	29.384	55.751	0.000	↓↓
ApEn (-)	0.856	0.086	0.796	0.893	0.809	0.153	0.692	0.929	-	-
SampEn (-)	1.311	0.337	1.128	1.502	0.952	0.405	0.663	1.210	0.000	↓↓
D2 (-)	2.603	1.018	1.798	3.148	1.187	1.241	0.320	2.765	0.002	↓↓
dfa1 (-)	1.419	0.219	1.346	1.590	0.988	0.482	0.656	1.542	0.000	↓↓
dfa2 (-)	0.651	0.220	0.572	0.864	0.690	0.213	0.592	0.847	0.549	↑
RPImean (beats)	9.821	3.798	8.565	12.312	12.828	8.599	9.998	15.587	0.021	↑↑
RPImax (beats)	135.000	48.793	82.000	156.000	109.000	72.475	62.000	181.000	0.734	↓
REC (%)	31.491	8.596	26.695	38.403	40.450	15.503	31.421	49.258	0.011	↑↑
RPadet (%)	98.780	1.003	98.046	99.207	99.154	1.797	98.224	99.685	0.089	↑
ShanEn (-)	3.008	0.307	2.854	3.225	3.232	0.429	3.030	3.506	0.021	↑↑

MD.: Median; SD: Standard Deviation; trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) under stress ($p > 0.05$); -: not computable. In bold HRV features changing significantly between rest and stress conditions.

Table 5.5: HRV features in rest and stress from 1 min NN data series. AE experiment.

Ultra-short term: 1 min										
HRV features	Rest				Stress				p-value	Trend
	MD	SD	25 th	75 th	MD	SD	25 th	75 th		
MeanNN (ms)	725.169	87.204	658.022	777.922	492.167	78.866	450.060	539.117	0.000	↓↓
StdNN (ms)	48.554	18.505	40.710	63.065	33.978	17.420	25.249	43.979	0.000	↓↓
MeanHR (1/min)	83.153	10.087	77.476	91.438	122.54	18.442	111.54	133.686	0.000	↑↑
StdHR (1/min)	5.799	2.433	4.624	6.731	8.003	3.500	6.124	10.981	0.000	↑↑
RMSSD (ms)	30.806	15.435	25.018	38.449	31.269	18.370	13.395	45.394	0.823	↑
NN50 (-)	7.500	8.396	4.000	14.000	11.500	14.956	1.000	23.000	0.382	↑
pNN50 (%)	10.201	10.539	4.444	17.722	10.156	11.444	0.848	16.556	0.505	↓
LF (ms ²)	1605.80	1567.53	563.291	2711.66	284.95	828.234	112.01	725.210	-	-
HF (ms ²)	367.511	381.504	190.97	497.14	103.37	280.790	42.947	230.933	0.000	↓↓
LF/HF (-)	4.398	3.216	2.906	5.730	3.443	4.408	1.427	6.237	-	-
TotPow (ms ²)	2557.33	2143.55	1464.61	4488.42	550.97	1614.007	212.826	1335.339	-	-
SD1 (ms)	21.919	10.988	17.790	27.369	22.197	13.042	9.514	32.244	0.775	↑
SD2 (ms)	66.060	24.540	55.730	85.111	36.032	24.563	27.547	53.614	0.000	↓↓
ApEn (-)	0.602	0.085	0.545	0.655	0.629	0.098	0.584	0.696	-	-
SampEn (-)	1.305	0.336	1.166	1.639	0.984	0.455	0.727	1.474	0.001	↓↓
D2 (-)	2.509	0.875	1.737	2.868	1.317	1.196	0.417	2.632	0.002	↓↓
dfa1 (-)	1.473	0.221	1.205	1.577	1.200	0.505	0.785	1.545	0.009	↓↓
dfa2 (-)	0.684	0.337	0.552	0.957	0.684	0.276	0.553	0.913	0.540	↓
RPImean (beats)	8.410	2.493	7.388	9.519	10.558	4.509	7.426	13.639	0.075	↑
RPImax (beats)	67.000	16.783	51.000	77.000	71.000	32.599	36.000	97.000	0.681	↑
REC (%)	29.679	7.495	24.992	36.131	34.787	13.899	25.887	44.335	0.107	↑
RPadet (%)	98.278	1.311	97.248	98.799	98.076	2.246	95.955	99.301	0.957	↓
ShanEn (-)	2.659	0.293	2.471	2.864	2.933	0.372	2.623	3.151	0.005	↑↑

MD.: Median; SD: Standard Deviation; trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) under stress ($p > 0.05$); -: not computable. In bold HRV features changing significantly between rest and stress conditions.

Table 5.6: HRV features in rest and stress from 30 sec NN data series. AE experiment.

Ultra-short term: 30 sec										
HRV features	Rest				Stress				p-value	Trend
	MD	SD	25 th	75 th	MD	SD	25 th	75 th		
MeanNN (ms)	720.853	89.963	650.870	772.564	480.738	72.726	444.000	529.298	0.000	↓↓
StdNN (ms)	48.619	18.875	33.701	66.484	31.894	18.916	19.215	41.428	0.000	↓↓
MeanHR (1/min)	83.864	10.562	77.769	92.424	124.951	18.788	113.421	135.618	0.000	↑↑
StdHR (1/min)	5.583	2.782	4.300	6.792	6.850	4.633	5.209	11.562	0.021	↑↑
RMSSD (ms)	30.829	17.345	21.260	42.370	28.998	21.791	10.732	47.465	0.247	↓
NN50 (-)	3.500	5.190	1.000	8.000	3.500	8.143	0.000	14.000	1.000	↑
pNN50 (%)	8.957	12.375	2.564	22.500	5.489	11.979	0.000	19.737	0.238	↓
LF (ms ²)	1207.038	1820.654	525.548	2956.652	180.888	1156.513	83.894	512.759	-	-
HF (ms ²)	242.283	928.954	142.342	453.410	69.902	214.315	31.608	157.711	-	-
LF/HF (-)	5.292	5.237	1.760	8.531	3.699	7.685	1.320	7.962	-	-
TotPow (ms ²)	2051.232	7451.549	1058.81	3868.496	366.116	2403.753	162.112	1198.807	-	-
SD1 (ms)	22.090	12.422	15.277	30.397	20.695	15.527	7.671	33.938	0.239	↓
SD2 (ms)	64.777	25.347	43.553	89.512	34.382	25.793	22.181	47.197	0.000	↓↓

MD.: Median; SD: Standard Deviation; trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) under stress ($p > 0.05$); -: not computable. In bold HRV features changing significantly between rest and stress conditions.

5.2.8.2 Multi-scale HRV comparison: short VS ultra-short

Table 5.7 summarises the results of the significance and trend analyses, presenting the HRV features' trends at each time-scale. Table 5.7 also reports which features were calculated for the different excerpt lengths (i.e., features indicated with '-' were not computable).

Table 5.7: HRV features' trends.

HRV features	5 min	3 min	2 min	1 min	30 sec
MeanNN (ms)	↓↓	↓↓	↓↓	↓↓	↓↓
StdNN (ms)	↓↓	↓↓	↓↓	↓↓	↓↓
MeanHR (1/min)	↑↑	↑↑	↑↑	↑↑	↑↑
StdHR (1/min)	↑↑	↑↑	↑↑	↑↑	↑↑
RMSSD (ms)	↑	↑	↑	↑	↓
NN50 (-)	↑	↑	↑	↑	↑
pNN50 (%)	↓	↓	↑	↓	↓
LF (ms ²)	↓↓	↓↓	↓↓	-	-
HF (ms ²)	↓↓	↓↓	↓↓	↓↓	-
LF/HF (-)	↓↓	↓↓	↓↓	-	-
TotPow (ms ²)	↓↓	↓↓	↓↓	-	-
SD1 (ms)	↑	↑	↑	↑	↓
SD2 (ms)	↓↓	↓↓	↓↓	↓↓	↓↓
ApEn (-)	↓↓	↓↓	-	-	-
SampEn (-)	↓↓	↓↓	↓↓	↓↓	-
D2 (-)	↓↓	↓↓	↓↓	↓↓	-
dfa1 (-)	↓↓	↓↓	↓↓	↓↓	-
dfa2 (-)	↑	↑	↑	↓	-
RPlmean (beats)	↑↑	↑↑	↑↑	↑	-
RPlmax (beats)	↓↓	↓↓	↓	↑	-
REC (%)	↑↑	↑↑	↑↑	↑	-
RPadet (%)	↑↑	↑	↑	↓	-
ShanEn (-)	↑↑	↑↑	↑↑	↑↑	-

MD.: Median; SD: Standard Deviation; trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) under stress ($p > 0.05$); -: not computable.

As shown in Table 5.7, from 5 min excerpts of NN data series, 18 of the 23 selected HRV features showed significant changes from the resting to the stress condition. 12 of these 18 features decreased significantly during stress, while the remaining 6 features (MeanHR, StdHR, RPl_{mean}, REC, RPadet and ShanEn), showed a significant increase. The second column in Table 5.7 demonstrates that from 3 min excerpts of NN data series all of the 23 features were computable and 12 features decreased significantly during stress, while 5 (MeanHR, StdHR, RPl_{mean}, REC, and ShanEn) increased significantly. However, RPadet, which showed significant increase during 5 min excerpts, failed to show any significant change when the data length was shortened below 5 min. The changes in the features extracted from the 2 min excerpts, shown in the third column of Table 5.7, present the same significant trends as the 3 min features, apart from ApEn, which is not computable, and RPl_{max}, which

is no longer significant ($p\text{-value}<0.05$). The changes in the features extracted from 1 min excerpts, shown in the fourth column of Table 5.7, present the same significant trends as the 2 min features, except for 3 HRV frequency features (LF, LF/HF ratio, TotPow), which are not computable, and 2 non-linear HRV features (RPI_{mean} and REC), which are no longer significant ($p\text{-value}<0.05$). The changes in the features extracted from the 30 sec excerpts, shown in the fifth column of Table 5.7, present the same significant trends as the 1 min features, apart from those features that are not computable.

Table 5.8 shows the results of the correlation analysis. Time domain HRV features maintained a significantly high correlation coefficient at 3 min, 2 min, and 1 min. Conversely, from 30 sec excerpts, StdNN showed a Spearman correlation coefficient above 0.70 at rest and below 0.70 during stress, while StdHR showed a Spearman coefficient below 0.70 during both rest and stress. Regarding frequency domain HRV features, they were highly correlated with the equivalent short HRV features at each time-scale (i.e., from 3 min to 1 min) during both rest and stress conditions. As far as non-linear features are concerned, SD1 maintained a constant behaviour between the short and ultra-short term excerpts during the rest and stress sessions whereas SD2 was less correlated over 30 sec during stress. ApEn, SampEn, D2, RPI_{mean} , RPI_{max} , REC, RPadet and ShanEn were highly correlated with short term HRV features for the 3 min excerpts during the rest and stress conditions, but they resulted in being less correlated in shorter time-scales. In general, HRV features resulted less correlated in rest than during stress conditions. This is most likely due to the fact that HRV showed a more depressed dynamic during stress. Due to this first analysis, the HRV features computed on 30 sec excerpts were at this point excluded from the rest of the study due to the low number of HRV features behaving coherently with the benchmark.

The results from the correlation analysis were supported by the visual inspection of the Bland-Altman plots. A decrease in the bias and the 95% LoA was observed as the excerpt lengths increased for all of the HRV features (see Appendix B, section B.1, Fig. B.1, B.2, B.3, B.4, B.5, B.6).

As a result, MeanNN, StdNN, MeanHR, StdHR, HF and SD2 were selected as good surrogates of short HRV features to detect mental stress, as they responded consistency across all the excerpt lengths (i.e., from 5 min to 1 min). Moreover, the discrimination power to automatically detect stress from these features across all the excerpt lengths (i.e., from 5 min to 1 min) was also proved as detailed in the next section.

Table 5.8: Correlation analysis of ultra-short HRV VS short HRV features.

HRV features	Rest Phase				Stress Phase			
	3 vs 5 min	2 vs 5 min	1 vs 5 min	30 sec vs 5 min	3 vs 5 min	2 vs 5 min	1 vs 5 min	30 sec vs 5 min
MeanNN (ms)	0.984	0.890	0.975	0.936	0.985	0.937	0.955	0.964
StdNN (ms)	0.954	0.875	0.905	0.749	0.962	0.912	0.791	0.640
MeanHR (1/min)	0.984	0.891	0.975	0.947	0.985	0.938	0.954	0.964
StdHR (1/min)	0.914	0.789	0.796	0.635	0.971	0.904	0.784	0.696
RMSSD (ms)	0.961	0.914	0.946	0.859	0.983	0.928	0.915	0.852
NN50 (-)	0.972	0.883	0.949	0.822	0.971	0.920	0.905	0.894
pNN50 (%)	0.967	0.882	0.943	0.818	0.969	0.915	0.913	0.881
LF (ms ²)	0.894	0.886	-	-	0.921	0.916	-	-
HF (ms ²)	0.915	0.906	0.901	-	0.925	0.915	0.798	-
LF/HF (-)	0.830	0.839	-	-	0.846	0.807	-	-
TotPow (ms ²)	0.897	0.882	-	-	0.900	0.905	-	-
SD1 (ms)	0.961	0.914	0.945	0.862	0.983	0.928	0.915	0.852
SD2 (ms)	0.956	0.865	0.876	0.707	0.941	0.898	0.755	0.694
ApEn (-)	0.771	0.169	-	-	0.918	0.790	-	-
SampEn (-)	0.855	0.666	0.681	-	0.931	0.826	0.599	-
D2 (-)	0.922	0.674	0.330	-	0.967	0.876	0.816	-
dfa1 (-)	0.661	0.687	0.637	-	0.927	0.908	0.799	-
dfa2 (-)	0.633	0.611	0.673	-	0.767	0.563	0.485	-
RPlmean (beats)	0.837	0.708	0.645	-	0.901	0.730	0.503	-
RPlmax (beats)	0.738	0.588	0.583	-	0.896	0.737	0.678	-
REC (%)	0.880	0.643	0.608	-	0.892	0.689	0.513	-
RPadet (%)	0.852	0.645	0.495	-	0.948	0.817	0.642	-
ShanEn (-)	0.795	0.661	0.614	-	0.907	0.720	0.463	-

All the correlations resulted significant ($p_\rho < 0.05$); in bold Spearman's correlation coefficient (ρ) greater than 0.7; -: not computable.

5.2.8.3 Classification and performance measurements

Regarding the feature selection process, all of the six HRV features (MeanNN, StdNN, MeanHR, StdHR, HF, and SD2), selected as good surrogates of short HRV features resulted as also being relevant in Folder 1. This was not a trivial result given the lower number of subjects included in Folder 1. In fact, a reduction in the number of subjects may result in an increase in the p-values. Among the 6 features, as shown in Table 5.9, SD2 resulted in being significantly correlated with 4 features, therefore, it was the first excluded. MeanNN resulted in being highly correlated to MeanHR, as expected, but not with the other features. Between MeanNN and MeanHR, MeanNN was chosen as it is easier to compute using wearable devices. Regarding the two standard deviations, StdHR was selected as it was not correlated with the other features. Consequently, HF, which resulted as being significantly correlated only with StdNN (which was excluded), was selected too. Therefore, as result of the redundancy analysis, the minimum set of relevant but mutually non-correlated features resulted to be: MeanNN, StdHR, and HF.

Table 5.9: Correlation among HRV features in Folder 1.

	MeanNN	StdNN	MeanHR	StdHR	HF	SD2
MeanNN	1	0.669	-0.980	-0.518	0.503	0.755
StdNN		1	-0.614	0.205	0.815	0.979
MeanHR			1	0.591	-0.454	-0.720
StdHR				1	0.138	0.038
HF					1	0.785
SD2						1

All the correlations resulted as being significant ($p_\rho < 0.05$); in bold the Spearman's correlation coefficient (ρ) greater than 0.7.

Each machine learning method was trained and validated with this combination of short HRV features using Folder' 1. The classifiers were then tested on short HRV features using Folder 2 as shown in Table 5.10.

Table 5.10: Model performance measurements estimated on the test set (Folder 2) for 5 min excerpts.

Meth.	Parameters	AUC	SEN	SPE	ACC
MLP	LR=0.3; M=0.2; NE=500	98%	100%	88%	94%
SVM	PolyKernel, E=1.0	88%	88%	88%	88%
C4.5	CF=0.25; ML=2	94%	88%	100%	94%
IBK	K=2	99%	88%	100%	94%
LDA	-	98%	88%	100%	94%

Meth.: methods; MLP: Multilayer Perceptron; SVM: Support Vector Machine; C4.5: decision trees; IBK: Neighbor Search; LDA: Linear Discriminate Analysis; LR: Learning Rate; M: Momentum; NE= Number of Excerpts; E=Degree; CF= Confidence Factor; ML= Minimum Number of Instances per Leaf; AUC: area under the curve; SEN: sensitivity; SPE: specificity; ACC: accuracy.

According to the criteria defined in Chapter 4 and section 2.4.2, the IBK classifier showed the highest AUC with 88% sensitivity, 100% specificity, 94% accuracy, and 99% AUC, using MeanNN, StdHR and HF as HRV features. Therefore, the IBK was chosen as the model to automatically detect mental stress. The IBK model was then tested using ultra-short HRV features in Folder 2 to further evaluate their capability to automatically detect mental stress (Table 5.11).

Table 5.11: Model performance measurements on different time-scale excerpts.

Duration	AUC	SEN	SPE	ACC
3 min	97%	94%	94%	94%
2 min	93%	94%	88%	91%
1 min	93%	82%	94%	88%

AUC: area under the curve; SEN: sensitivity; SPE: specificity; ACC: accuracy.

The ROC curves for the final model at different lengths are presented in Fig. 5.8.

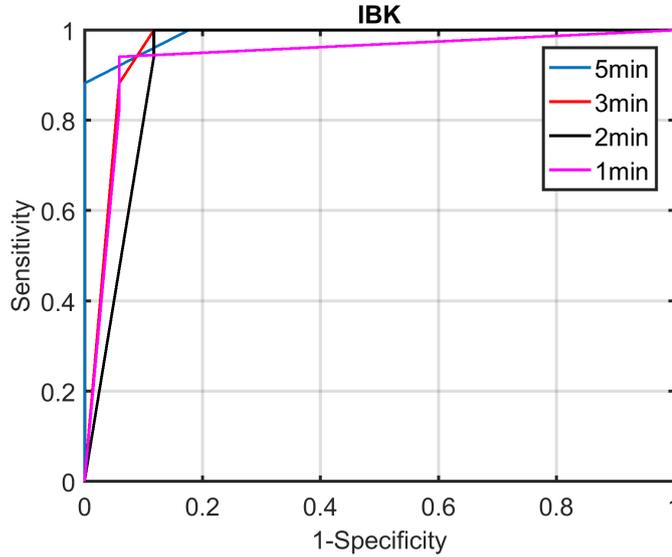


Figure 5.8: ROC curves of the IBK final model developed using real-life data for different time-scale excerpts.

The length of the data seemed to affect the performance of the model to a small degree. However, as shown in Table 5.11, the model outperformed for 3 min time-scale with 97% AUC. In fact, compared to the short term performances, using 3 min excerpts sensitivity increased by 6% and specificity decreased by 6% respectively. Nevertheless, the model still achieved good performances also using 1 min HRV excerpts. After observing these results, the model was also assessed on all consecutive 1 min excerpts (as shown in Fig. 5.6, right-hand side) within the 5 min NN data series in order to see if the performances changed based on the extracted excerpts. The performances using the 1 min HRV features proved to be consistently good with $86 \pm 4.1\%$ sensitivity, $95 \pm 4.4\%$ specificity and $92 \pm 3.75\%$ accuracy.

5.2.9 Discussion

The current experiment aimed to investigate if ultra-short HRV features were reliable surrogates of short term ones to automatically detect mental stress. This is a topic of growing interest as demonstrated by the numerous apps and wearable devices presented on the market. However, as demonstrated in the previous Chapter 3, to the best of the researcher's knowledge, none of the studies in the existing literature investigated and identified in a rigorous way any subset of ultra-short features to automatically detect mental stress.

Regarding the method, this study presented an innovative framework to assess the minimum length of HRV excerpts to detect mental stress in healthy young subjects. In fact, only two studies [160, 161] evaluated the reliability of ultra-short HRV features during a stress condition. As reported in Chapter 3, section 3.2.3, in Tables 3.7 and 3.7a, Pereira *et al.* [160] used only a parametric statistical test (one-way ANOVA) to determine which HRV features (i.e., 220, 150, 100 and 50 sec) could discriminate between rest and stress sessions (p -value <0.05) with small windows of analysis. The results from Pereira *et al.* [160] showed that MeanNN, StdNN, RMSSD and pNN20 are the most discriminating features to differentiate between stress in 50 sec window; in the presented study, MeanNN and StdNN over 60 sec were also considered as good surrogates of short ones, whereas RMSSD showed that has not able to discriminate between rest and stress and pNN20 was not computed according to the existing guidelines [15]. Salahuddin *et al.* [161] used only the non-parametric Kruskal Wallis test to assess that ultra-short term analysis was not significantly different to short term analysis if the p -value was greater than 0.05 and Wilcoxon sign-rank test (p -value <0.05) to find the shortest duration that distinguished between rest and stress. In other words, no correlation or machine learning methods were utilised to validate their findings. Moreover, if the p -value is greater than 0.05 then the null hypothesis can be neither rejected nor accepted. Therefore, no conclusions can be drawn using only the statistical significance tests (e.g., Kruskal Wallis test), which make the results reported in [161] not sufficiently reliable. In addition, the Salahuddin *et al.* study enrolled only 24 subjects. Therefore, even if the method proposed in [161] could be considered reliable (and it is not), a higher number of subjects would have resulted in different (probably smaller) p -values, changing the results entirely. Unfortunately, this study [161] was the paper used as justification by the majority of works in this area for the use of ultra-short HRV features. In fact, many wearable systems [140, 152, 154] and scientific studies [147, 148, 153, 164] monitoring stress via ultra-short term HRV analysis have based their feature selection on Salahuddin *et al.* [161] results.

Two other studies [151, 157] investigated the use of ultra-short HRV features, but in different conditions (i.e., physical activity, acoustic stimulation). In fact, Esco and Flatt [151] determined the agreement between ultra-short RMSSD feature computed in 10, 30 and 60 sec and conventional longer excerpts of 5 min in male athletes under resting and post-exercise conditions. They used ANOVA test to investigate if that ultra-short RMSSD was not significantly different from the standard 5 min RMSSD ($p\text{-value} > 0.05$) and used intra-class correlation coefficient (ICC) analysis to investigate if ultra-short RMSSD was highly correlated with 5 min RMSSD. However, no statistical tests were run to evaluate the significant difference and trends of RMSSD were not investigated between pre- and post-exercise. Lastly, Nardelli *et al.* [157] investigated the reliability of SD1 and SD2 from 15 to 60 sec compared to 5 min and 1 hour in healthy subjects at rest and during an experiment of emotional acoustic simulation. They used Spearman's correlation analysis and Bland-Altman plots. However, they only investigated two non-linear features and no statistical tests were run (or reported) to evaluate the significant difference and trends in SD1 and SD2 between rest and effective sounds.

Moreover, few studies have assessed the reliability of ultra-short HRV features as surrogates of standard short HRV features only during controlled resting conditions. Nussinovitch *et al.* [158] compared HRV calculated from 5 min, 1 min and 10 sec excerpts from 70 subjects using intra-class correlation (ICC), which is a parametric correlation test assuming that variables are normally distributed, which for frequency domain HRV features is never true unless a log-transformation is performed. McNames and Aboy [155] also used the ICC to compare the accuracy of HRV calculated from data lengths spanning from 10 sec to 10 min with 5 min recordings, using the R-R interval dataset posted on PhysioNet. Baek *et al.* [146] investigated the relationship between standard 5 min and ultra-short HRV, from a wide range of age groups under a controlled resting condition. They used Pearson's correlation to investigate the linear relationship between short HRV and ultra-short HRV, as well as the Kruskal-Wallis to test the statistical significance ($p\text{-value} > 0.05$) of differences between ultra-short and standard 5 min HRV. However, the Pearson's correlation would assume a normal distribution for both variables, but HRV features have shown not to be. Moreover, even if the correlation is high, if the Kruskal-Wallis $p\text{-value}$ is greater than 0.05 then the null hypothesis can be neither rejected nor accepted. Munoz *et al.* [156] investigated HRV features in the time domain in 10, 30 and 120 sec compared to 5 min, using a more rigorous method. In fact, they used Pearson's correlation, after log-transforming HRV features, Bland-Altman plots and Cohen's d . However, they investigated only two time HRV features (StdNN and

RMSSD) over a resting condition.

Regarding the results achieved in this study, the statistical analysis in the short term showed a significantly depressed HRV during stress, in agreement with the systematic literature review (Chapter 3, section 3.2.2). Ultra-short term HRV features also resulted in being significantly depressed during mental stress over each time-scale. Concerning the HRV features in time domain, all of them maintain the same behaviour across the 5 different time-scales (i.e., 5 min, 3 min, 2 min, 1 min, and 30 sec). Moreover, four of them (MeanNN, StdNN, MeanHR and StdHR) were also significantly different between rest and stress periods and were significantly correlated (Spearman's rank $\rho > 0.7$) across time-scales (i.e., each ultra-short vs short time-scale per each feature). These results, achieved with a more robust method, confirmed the findings of Baek *et al.* [146], McNames and Aboy [155], Nussinovitch *et al.* [158] and Munoz *et al.* [156], which showed that MeanNN, StdNN, MeanHR are reliable for lengths from 5 min to 1 min in controlled resting condition. However, some HRV features that showed to be good surrogates in the literature, failed to show good results in the present study. The interpretation of this result could be that the method used in the present study is based on more stringent and reliable requirements (i.e., inter-group and intra-group assessments through the use of appropriate statistical tests), compared to other studies, which demonstrated significant methodological limitations, as discussed above. Concerning the HRV features in the frequency domain, it is well-known that a minimum of 1 minute is required to estimate HF and a minimum of 2 minutes is required to estimate the LF component [15]. Accordingly, the present study proved that for HRV features in the frequency domain, such as LF, the minimum length is 2 minutes. Additionally, the HF component could be extracted for 1 min excerpts, as confirmed by the fact that in this study HF resulted in being a good surrogate of the 5 min equivalent. In fact, as also proved by Baek *et al.* [146], LF had a very low Pearson's coefficient below 2 min whilst HF below 1 min. In relation to non-linear HRV features, no study has investigated their reliability in excerpts shorter than 5 minutes. The current study empirically proved that non-linear HRV features lose their utility for excerpts below 3 minutes mainly due to computational problems. In fact, non-linear HRV features require a high number of samples in order to appreciate the dynamics of the heartbeat series over time. Only two non-linear HRV features (SD1 and SD2) showed to be good surrogates over 3, 2 and 1 min length as also shown by Nardelli *et al.* [157]. Therefore, it is recommended that researchers consider excerpt lengths above 3 minutes if interested in understanding how HRV reflects a chaotic system.

Lastly, the present study showed a model able to detect stress with higher

accuracy than the models presented in the existing literature. In fact, as also described in Chapter 3, section 3.2.2.12, two studies [45, 128] proposed a model to detect mental stress using short term HRV analysis.

Melillo *et al.* [45] adopted the same dataset as in this study. They proposed a model based on LDA, employing only three HRV non-linear features: SD1, SD2 and ApEn for short term HRV analysis (5 min). The model proposed in their study achieved sensitivity, specificity and accuracy, respectively of 86%, 95% and 90%, which are lower than the performance achieved by the model developed in this study. A possible reason for that may lie in the use of a less robust method to cope with small dataset. In fact, they did not perform a feature selection process on an independent folder and the use of linear classifier -such as LDA- may not be able to discriminate at high accuracy between rest and stress conditions. Another study conducted by Traina *et al.* [128] studied the Pearson's correlation among frequency domain measures before and after the stress session, demonstrating that those correlations were significant. However, as discussed above, the Pearson's correlation is based on the assumption that the HRV measures are normally distributed, but HRV frequency features are not.

Seven studies, as described in Chapter 3, section 3.2.3, Table 3.8, developed a model to detect mental stress using ultra-short HRV features. Mayya *et al.* [140] proposed a method for automatically detecting mental stress using smartphone and 1 min HRV features. The model was built on the assumption that ultra-short HRV features were relevant according to the available literature [161, 162], which has been proved to lack a robust method to identify ultra-short HRV features that are good surrogates of short HRV features. They used a multinomial logistic regression applied to 2 features, RMSSD and dfa1, and achieved 80.5% accuracy, which is lower than the accuracy achieved in the present study, supporting the idea that an erroneous ultra-short feature selection can generate low performances. Choi *et al.* [149], Brisinda *et al.* [148] and Sun *et al.* [164] also proposed a method to automatically detect mental stress focusing on 4 min, 2 min and 1 min HRV features respectively. Also in these studies, the models were built on the assumption that ultra-short HRV features were relevant according to the available literature, although Brisinda *et al.* confirmed their findings using only ICC analysis. These studies used linear classifiers achieving accuracy lower than the one achieved in the current study. Other models were developed using ultra-short term HRV analysis along with other physiological measurements, but are not discussed, as this is not in line with the scope of this study. Overall, none of those papers achieved better results than the ones presented in this study. This also supports the fact that a

reliable identification of good surrogates is important in order to identify a good set of features aiming to detect mental stress. The current study proved that IBK was able to detect stressed subjects with 88%, 100%, 94% sensitivity, specificity and accuracy respectively, using short term HRV features (MeanNN, StdHR and HF). IBK was the most recurrent machine learning used among the 7 papers identified in the existing literature [149, 159, 166].

5.2.10 Conclusion and applications

Currently, 5 min recordings are regarded as being an appropriate option for HRV analysis to detect mental stress in healthy subjects. However, the continued rise in the interest of everyday wearable devices being able to instantaneously assess mental stress level is rising the attention of the scientific community around the use of HRV features computed in excerpts shorter than 5 min.

This study proved that not all the ultra-short HRV features were good surrogates of short term ones. In fact, only six ultra-short HRV features resulted in being good surrogates of short term ones: MeanNN, StdNN, MeanHR, StdHR, HF, and SD2. These six features displayed consistency across all of the excerpt lengths (i.e., from 5 min to 1 min) and good performance if employed in a well dimensioned automatic classifier (i.e., each predictor variable presents at least 10 “occurrences” [76]). Moreover, an automatic classifier based on IBK was able to detect stressed subjects with very high performances, using 3 min HRV analysis, and relatively good performances using 1 min HRV excerpts. The former achieved sensitivity, specificity and accuracy of 94%, 94% and 94% respectively and the latter achieved 82% sensitivity, 94% specificity and 88% accuracy. The method employed to develop the machine learning model, described in Chapter 4, section 4.2.2.1, led to better results compared to the study conducted on the same dataset by Melillo *et al.* [45], which did not adopt different folders for training and testing and performed the feature selection process on all of the dataset.

In conclusion, it is possible to automatically detect mental stress using ultra-short HRV features with excerpts not shorter than 1 min. According to the specific application, 3 or 2 min excerpts could be preferable, because features having a clear physiological significance (e.g., HF and LF) remain computable. Finally, it is useful to mention that the proposed methodology could be used in any application aiming to automatically detect a condition using ultra-short HRV features. In particular, the proposed method can improve the identification of the minimal length of HRV excerpts enabling the detection of an anomaly in real time.

The results of the present work could be applied to the situation of a mental

effort such as an exam or job interview, that represents a long period under stress, but more studies are necessary to understand the response of HRV metrics in shorter fright situations.

Therefore, due to the low number of subjects enrolled in this study, more experimental studies were carried out to enrol more subjects in order to verify if ultra-short HRV feature may be useful to detect stress in a wider sample. The choice of having laboratory experiments was due to the degree of control it provides in order to assess the relationship between acute mental stress and ultra-short term HRV analysis. However, the major disadvantage of experimental studies is that the nature of the experiment may be very unlike what people might actually do in everyday life. Although the experiments were modelled as far as possible on simulating real acute mental stress according to the existing literature, the effect of in-lab stressors resulted in a less powerful stress; also this point is discussed in the following sections.

5.3 Detection of mental stress in laboratory settings

The study carried out on a real-life stressor (i.e., an Academic Examination (AE)) assessed the validity of ultra-short HRV features through a robust framework. However, due to the low number of subjects enrolled using the real-life stressor, experimental data were gathered through two independent experiments in-lab settings (deliverable 1d), by using the Stroop Colour Word Test (SCWT) and a highly paced video game challenge (VGC) as cognitive stressors. Two different in-lab stressors were used to investigate in more detail the effect of in-lab stress.

These studies aimed to explore the power of in-lab stressors (deliverable 1e) and to train, validate, and test an automatic model to detect mental stress using ultra-short HRV features in order to verify the relevance of using ultra-short HRV features to detect stress (deliverable 1f).

5.3.1 Experiment designs

Based on the results of the systematic literature review presented in Chapter 3, section 3.2.2, the main steps followed to design reliable and valid experiments were:

- identifying the population, the universe of people to which the study could be generalised;
- identifying a sample from the subset of the population;
- identifying the sampling frame, i.e., eligible members from the population;

- identifying the medical devices and instruments to use during the experiment;
- identifying the location of the experiment;
- identifying the protocol to follow during the experiments.

For both of the experiments (E2 and E3) every participant was exposed to the same environment, including the characteristics of the room and the instruction the participants received.

5.3.1.1 Study population

The minimum number of subjects was estimated using two different methods. The first method is the traditional statistical method. In fact, sample size is often a statistical concept that involves determining the number of observations or replicates (the repetition of an experimental condition used to estimate the variability of a phenomenon) that should be included in a statistical sample to have sufficient statistical power [264]. However, this concept is not completely true for predictive modelling. Therefore, a second method was used to estimate the minimum number of observations for predictive models.

The first method estimated a minimum number of 42 subjects. This was calculated using standard statistical methods [264]. The most common feature reported by previous studies was the power spectrum high frequencies (HF); the mean differences and the standard deviation of the HF were calculated during the meta-analysis conducted in Chapter 3, section 3.2.2. Therefore, since the experiments were repeated in the same subject twice under different conditions (i.e., rest and stress), the minimum number was calculated as follows: the mean HF difference in literature was 265.616 ms^2 with an estimated pooled standard deviation of $1,555 \text{ ms}^2$; assuming a type I error=0.05 and a power of 80% [264], the estimated minimum number was 42 subjects.

However, 42 subjects represent the minimum number for statistical purposes, therefore, according to Forest *et al.* [76], the minimum number for predictive modelling was estimated to be 150 subjects for the two independent experiments using SCWT and VGC. This number was calculated linking the model complexity to the population sampling. According to Foster [76], in order to avoid overfitting, a minimum number of 10 “events” is necessary for each attribute to result in a classifier with reasonable predictive value. For instance, an automatic classifier with a minimum of 3 unknown attributes (e.g., parameters and/or variables) should be trained using at least 30 observations. Moreover, in order to reduce the observations' interdependency, the classifier should be trained on only a third of the whole dataset, in

order to perform feature selection and testing on independent data. Therefore, the subjects should be split into three folders to reduce bias and overfitting problems [76, 265]:

- Folder 1 for feature selection;
- Folder 2 for training and validation;
- Folder 3 for testing.

Therefore, a minimum number of 30 subjects should be included in each folder.

However, the splitting of the subjects into folders for balanced datasets could be:

- 20% for Folder 1 (i.e., feature selection);
- 60% Folder 2 (i.e., training and validation);
- the remaining 20% in Folder 3 (i.e., testing).

Consequently, assuming that the minimum number of attributes would be 3, the number was estimated to be 150 subjects (Fig. 5.9).

Hence, the minimum number of subjects enrolled for the SCWT and VGC ranged from 42 to 150 volunteers.

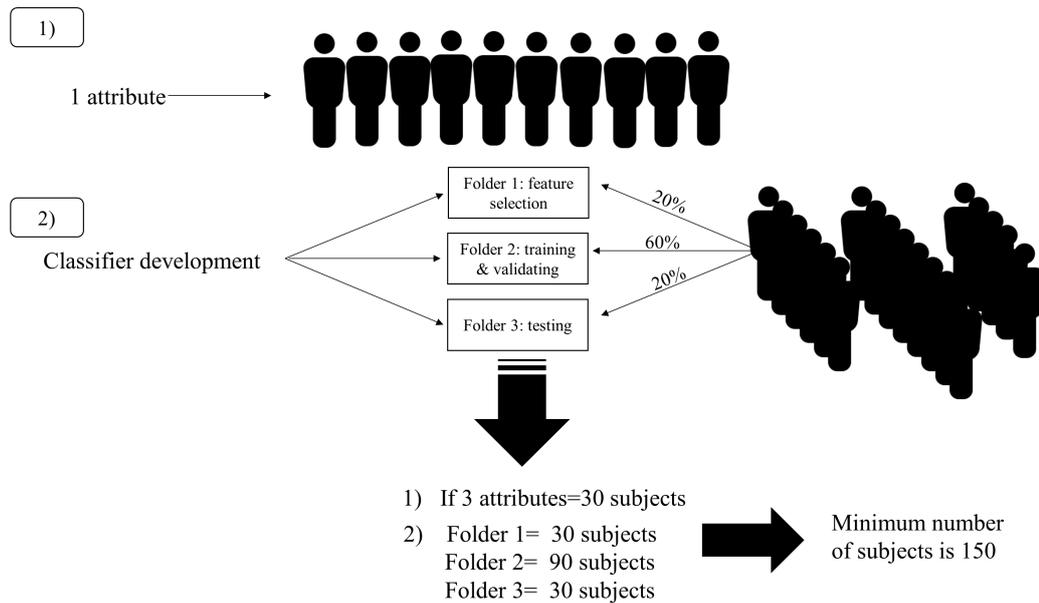


Figure 5.9: The minimum sample for predictive modelling. This method links the model complexity to the required population sampling. As rule of thumb, 10 observations (subjects) are needed for each attribute in the final model. Moreover, in order to reduce the observations' interdependency, the classifier should be trained only on a third of the whole dataset to perform feature selection and tested on independent data. Therefore, the subjects should be split into three folders, each of which with a specific percentage of observations.

To achieve a high degree of reliability and consistency, the criteria for eligibility of the sample were:

- age between 20 and 40 years old;
- no history of heart disease or systemic hypertension;
- normal BMI, i.e., 18.5 to 24.9;
- no consumption of drugs or alcohol before the experiment;
- for females, the experiment was carried out outside their menstrual cycle.

The previous criteria were chosen in agreement with the existing literature (Chapter 2, section 2.2.1.2) in order to generalise the results coming from the experiments. In fact, different studies have reported that ageing is associated with depressed HRV and parasympathetic function [266, 267]. As far as the second criterion is concerned, it has been demonstrated that hypertension leads to changes in HRV as well as heart disease may corrupt the ECG signal, entailing a lack of reliability in

the study [268, 269]. Moreover, a normal BMI avoids undesired corruptions of the signal and assures reliability and accuracy of the results [270, 271]. The condition that drugs and alcohol were forbidden before the experiment was made in order to not alter HRV patterns, since the consumption of drugs and alcohol varies the sympathetic and parasympathetic functions [272, 273]. Lastly, the menstrual cycle has also been shown to be a relevant measure for HRV features [254].

5.3.1.2 Stroop Word Colour Test (SCWT) and Video Game Challenge (VGC) implementation

According to the existing literature, the SCWT and the VGC were chosen as the two most common cognitive stressors used to study human psycho-physiological responses to mental stress in laboratory scenarios.

Cognitive stressors also called acute stressors induce cognitive demands [274]. The cognitive tasks also represent the most convenient and applicable acute stressors with a high psychological relevance to respondents [275]. In fact, they are:

- convenient, unobtrusive modes of test administration;
- applicable to a wide range of potential respondents;
- psychologically relevant to respondents;
- minimal, standardised motor response requirements.

Furthermore, sympathetic nerve response to mental stress is strongly influenced by the perception of difficult and demanding tasks such as the SCWT and the VGC.

In fact, they both induce changes in autonomic responses of the SNS, which relates to physiological measures, such as changes in HRV [13, 276, 277]. Different studies have shown that pacing a SCWT or VGC resulted in substantial heart rate accelerations [13, 276, 277]. In fact, they are accompanied by heightened HR levels and a decrease in MeanNN during the performance of a SCWT and a VGC. Moreover, the SNS indicator calculated by spectral analysis (LF, HF, LF/HF) of HRV might be sensitive to reflect a slight increase in cardiac SNS activity during a SCWT and a VGC, as shown in Table 3.4 [122, 129].

Stroop colour word test A SCWT is a challenging task, whose prominent feature is the conflict or interference situation during which the subject must name the colour of the ink of the ink-words when the colour and word are incongruent [13]. According to previous literature [278–281], the SCWT was designed in two parts in order to increase the level of difficulty over time.

The first part of the test showed congruent word-colour cards. The words were shown on a white background; the size of each card was 5 by 5 words. The first part lasted for 23 cards; each card for around 5 seconds. The subject was asked to read aloud as many words as he/she could, from left to right, without caring about the ink used for the words (Fig. 5.10, left-hand side).

The second part was designed employing incongruent cards at higher speed (i.e., 100 cards for 3 sec each). The cards lasted for the same time and they were skipped with the same speed, but the task of the subject was different. The subject was asked to pronounce the colour of the word-ink from left to right (Fig. 5.10, right-hand side).

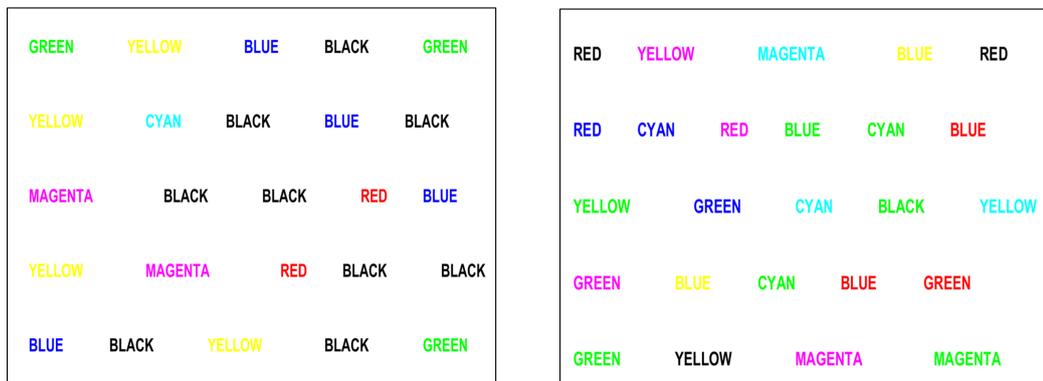


Figure 5.10: SCWT example slides. On the left-hand side, a congruent slide example (first part), during which the subject was asked to read aloud as many words as he/she could, from left to right, without caring about the ink of the words. On the right-hand side, an incongruent slide example (second part), during which the subject was asked to pronounce the colour of the word-ink from left to right.

A Matlab code producing congruent and incongruent colour-words with blue, green, black, red, yellow, magenta, cyan colours, was employed to generate four different kinds of cards with increasing difficulty level for the second part of the SCWT [282]:

- small problems with matching colours;
- large problems with matching colours;
- small problems with non-matching colours;
- large problems with non-matching colours.

The SCWT test was, then, assembled using video-maker software. A random combination of the four types of card was used in order to not create a repeated pattern

for the second part of the SCWT. Explanation slides were inserted before each part. Between the first and the second part a break of 24 seconds, with the relative explanations and examples, was included. The entire task lasted for not more than 7 minutes (Fig. 5.12).

Video game challenge A platform based on a commercially available first-person shooting game was created to mimic a situation of a hostage rescued by a special weapons and tactics counter-terrorist team to simulate a cognitive demanding task such as real working life stress. The game was modified in several ways by a software developer. The game was manipulated so that participants could not “die” to complete the task, they could use the same weapon (i.e., a gun) or change weapon (i.e., knife). The task was always performed under the same scenario (i.e., the same game map) with the hostage position kept constant. Subjects were unaware of the manipulation of the commercial game. The entire experiment was remotely run and monitored through a centralised server. This also allowed us to provide detailed instructions to the participants at the beginning of the task.

For the first 3 minutes of the session and before the beginning the experimental task, the author demonstrated the main commands to play the game and participants performed a round trial of 1 minute to familiarise themselves with the controls. The game was ranked as adequate for subjects aged 16 and above. The video game interface is shown in Fig. 5.11.



Figure 5.11: VGC interface.

5.3.1.3 Ethical approvals

These studies were approved by the Biomedical and Scientific Research Ethics Committee (BSREC) of The University of Warwick, assuring the anonymity and no side-effects or possible disadvantages for the subjects. All subjects were carefully instructed on the study protocol and informed consent was given prior to examinations (ref. REGO-2014-656, REGO-2014-656 AMO1). BSREC approval letters are

reported in Appendix B, section B.2.

5.3.1.4 Study protocol

All subjects were examined under standard conditions: a quiet room with minimisation of stimuli, during the morning, minimising physical motions and other stimuli possibly affecting HRV.

5.3.1.5 SCWT protocol

128 healthy volunteers with no history of heart disease, systemic hypertension or another disease potentially influencing HRV were enrolled. They were not obese and did not consume medication, drugs or alcohol in the 24 hours preceding the experiment. All the subjects had normal or corrected-to-normal vision.

Volunteers were instructed to sit comfortably in an arm-chair and not move unless necessary. Continuous ECG recordings were performed during rest and stress sessions. The rest session was recorded for 6 minutes during which the subjects were asked simple questions (see Appendix B, section B.2), regarding age, weight and height in order to induce them to talk. Before the stress session started, a brief introduction to the test was explained through demonstrative videos. Finally, the SCWT was administered using a 28" screen for 7 nominal minutes.

ECGs were recorded continuously during the experiment. No ECG signal was recorded during the instructions.

(see Appendix B, section B.2) using a mouse and keyboard, in order to standardise the conditions between the rest and stress sessions (i.e., use of the PC). During the stress session, the shooter video-game containing fast-paced content (i.e. war scenes, gun fighting) was shown using a 24" screen, and ECG signal was recorded for the entire duration of the game, namely 5 minutes (Fig. 5.13).

The ECGs were recorded continuously during the experiment. No ECG signal was recorded during the instructions.

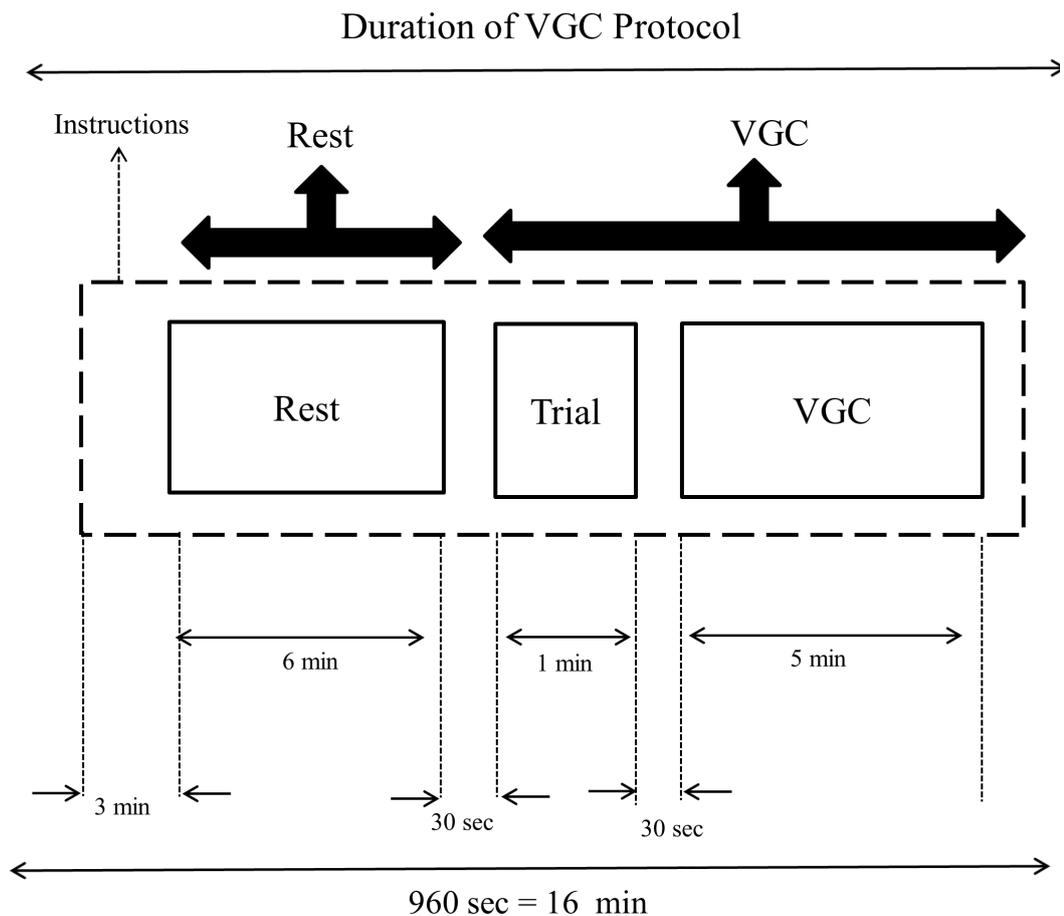


Figure 5.13: VGC protocol. At the start of the experiment, the author gave instructions (3 min) to the participants regarding the task that they needed to undertake. An ECG signal was acquired during resting condition for 6 nominal minutes during which the participants filled in a survey on demographic information. After a small break (approximately 30 sec), a trial of 1 min was performed, and finally, the participants played the VGC for 5 nominal minutes.

5.3.2 Hardware and software

Hardware For both experiments, the ECG signal was recorded using a Zephyr BioPatch device (Annapolis, USA) as shown in Fig. 5.14. The Zephyr BioPatch can be used in two different ways through the patch or through the strap, both ways expected to be positioned under clothing on the left under the chest. In respect to a subject's privacy, the subject was invited to go into a separate room to wear the device. To avoid any gender-related discomfort or embarrassment, the subject was assisted by a staff member of his/her gender.



Figure 5.14: Zephyr BioPatch (on the left-hand side) and Zephyr BioPatch side strap (on the right-hand side).

A one lead ECG was acquired with a sampling frequency of 250 Hz and a digital resolution up to 12 bits. The BioPatch is a unique and easy way to record ECG signals. It is comfortable, small and it attaches to traditional disposable ECG electrodes. Furthermore, Heart Rate Confidence and Respiration Confidence are continually calculated using the signal to noise ratio and other parameters that indicate the device is worn correctly and the data are medically valid. The ECG waveform is filtered and amplified based on proprietary circuitry designed for high levels of body movement. Data were continuously logged using an on-board flash memory. ECGs were subsequently downloaded to a PC for further offline pre-processing as explained in the next sections.

Moreover, in both experiments, the Nexus 10 was also used as benchmark. The NeXus-10-MK II (Mind Media, Herten, The Netherlands) offers 4 single channel inputs, 2 dual channel inputs, 1 oximetry/ trigger (with NeXus Trigger Interface) input and 1 digital input. The Nexus-10 is capable of measuring a wide variety of modalities simultaneously, such as brainwaves (EEG, SCP), muscle tension, heart rate, relative blood flow, skin conductance, respiration and temperature. It communicates wirelessly through Bluetooth or use of a USB extender cable, in order to record data at higher sample rates. It is equipped with a high-grade lithium-ion battery pack and an SD flash memory card slot, enabling full ambulatory use of this portable device. With the high medical grade connectors that lock-in, there is neither noise nor artefact when touching or pulling them. Artefacts through movement of cables or other external sources of noise are reduced to a minimum, because

of cutting-edge active noise cancellation technology and carbon coated cables. The single-lead ECG was acquired with a sampling frequency of 250 Hz and a digital resolution up to 24 bits. Data acquired with the Nexus 10 were used in case the signal acquired from the Zephyr BioPatch was corrupted. However, it was not the case for these experiments.



Figure 5.15: The NeXus-10 (MindMedia).

Software Different analyses were carried out using different software. The pre-processing of ECG signals was carried out using the PhysioNet's toolkit as detailed in section 5.2.2. HRV analysis was conducted using Kubios. A full description is detailed in 5.2.2. All of the statistical analysis were carried out using in-house tools developed in Matlab2016b. Machine learning algorithms were developed using the Weka Platform (version 3.8.01) and Matlab2016b software.

5.3.3 Data analysis

Fig. 5.16 describes the main stages of the data analysis carried out for this study. The acquired ECGs were analysed and HRV features extracted from short term excerpts (5 min) and 1 min excerpts, which resulted in the shortest length to reliably observe changes during stress, as detailed in section 5.2.8. The statistical analyses were run separately for the SCWT and the VGC. HRV features were investigated for 5 min and 1 min excerpts. The effect of real and in-lab stressors was also explored. Finally, the discrimination power of ultra-short HRV features in detecting mental stress was investigated via machine learning techniques.

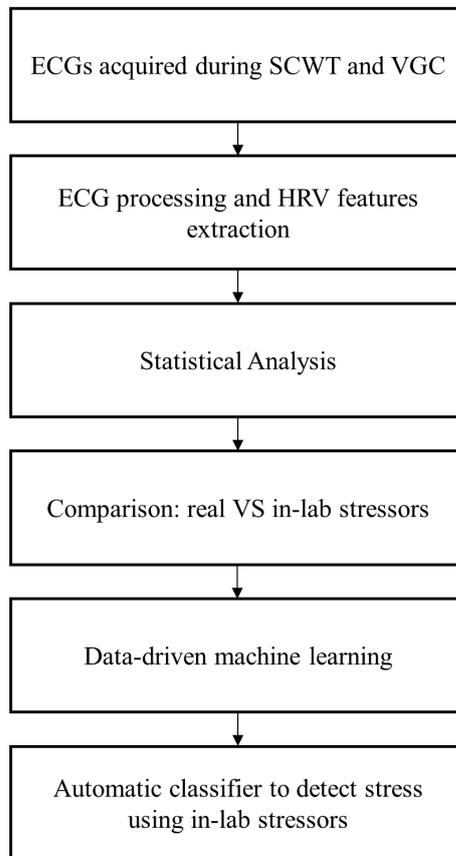
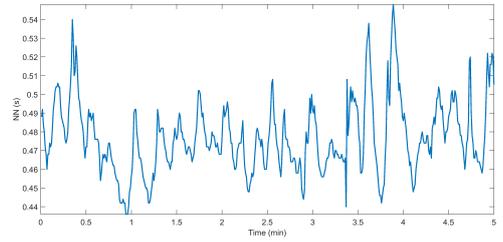
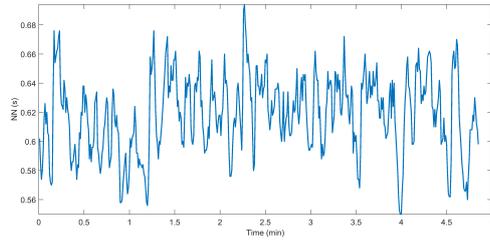


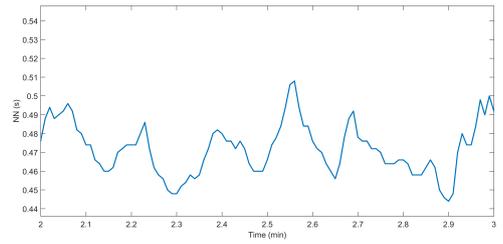
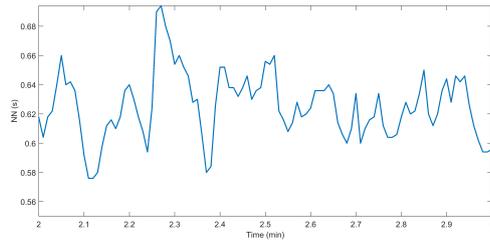
Figure 5.16: Data analysis flow for in-lab stress. ECGs were acquired during stressful situations, pre-processed and HRV features extracted. Statistical analysis identified HRV features that changed significantly during rest and stress conditions. The effect of real and in-lab stressors was investigated. Data-driven machine learning methods (i.e., SVM, MLP, IBK, C4.5 and LDA) were used to develop an automatic classifier to detect passive stress via ultra-short HRV features.

5.3.4 HRV analysis

As shown in Fig. 5.19, the RR interval time series were extracted from ECG records using an automatic QRS detector, WQRS, available in the PhysioNet's toolkit [255]. An illustrative example of the raw NN (or RR, since no ectopic beats were detected) interval series over different time scales is shown in Figs. 5.17 and 5.18 for the SCWT and VGC respectively. However, no conclusions can be drawn from these raw NN series, therefore, HRV features are then extracted and analysed.

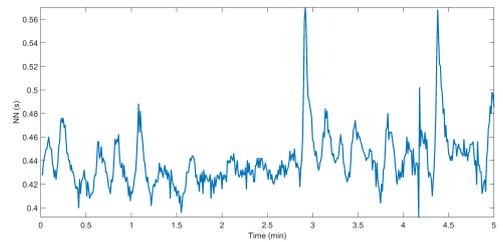
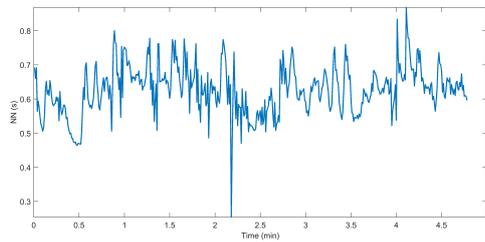


(a) Raw NN series at 5 min during rest session. (b) Raw NN series at 5 min during stress session.

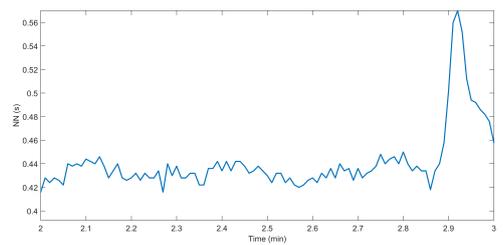
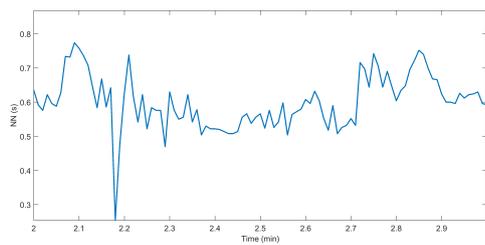


(c) Raw NN series at 1 min during rest session. (d) Raw NN series at 1 min during stress session.

Figure 5.17: Raw NN series for one subject during the SCWT rest and stress sessions over 5 min and 1 min excerpts.



(a) Raw NN series at 5 min during rest session. (b) Raw NN series at 5 min during stress session.



(c) Raw NN series at 1 min during rest session. (d) Raw NN series at 1 min during stress session.

Figure 5.18: Raw NN series for one subject during the VGC rest and stress sessions over 5 min and 1 min excerpts.

QRS review and correction were performed using WAVE. The automatic QRS detection was followed by manual review.

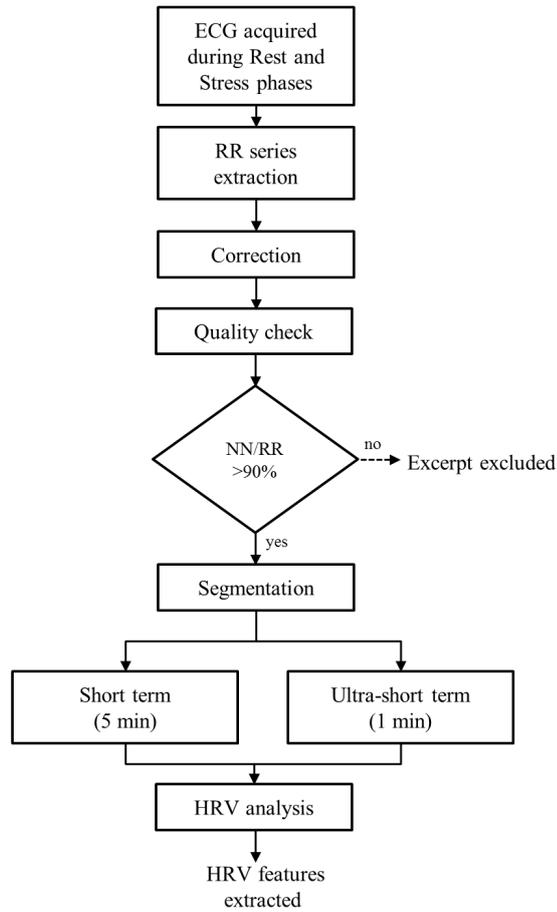
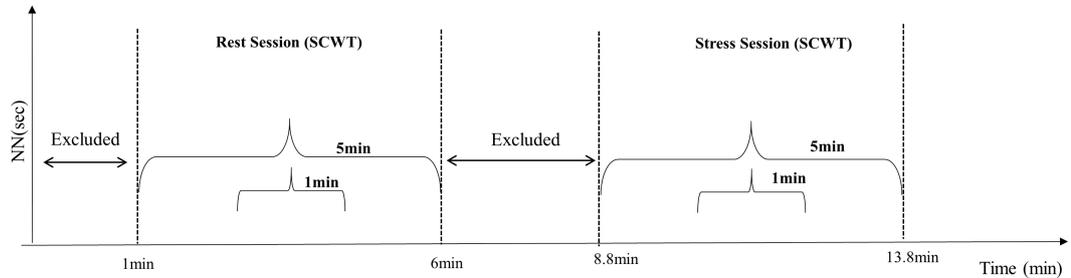


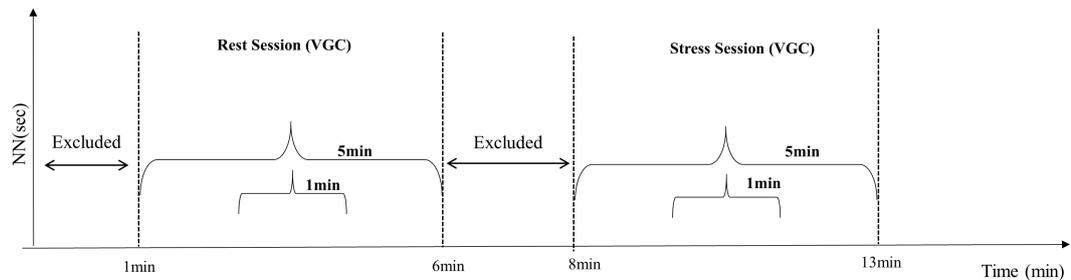
Figure 5.19: HRV processing workflow for in-lab stress. NN/RR is the ratio of the total RR intervals labelled as NN (normal-to-normal beats). Short term HRV is analysed in 5 min excerpts. Ultra-short term HRV is analysed in excerpts of 1 min length.

The fraction of total RR intervals labelled as normal-to-normal (NN) intervals was computed as NN/RR ratio. Thresholds of 80% [255] and 90% [258] have been proposed. In studies enrolling only healthy and young subjects, a lower NN/RR ratio is mainly associated with movement artefacts. In the current study, in which subjects were healthy and young, sitting in a comfortable position, a threshold of 90% was chosen and still no records were excluded. The ECGs from both experiments were segmented and the last 5 min were extracted from the rest and stress sessions and HRV features were computed as shown in Fig. 5.20. Moreover, the central 1 min segments (Fig. 5.20) in both rest and stress sessions and for both experiments were also extracted and HRV features were computed. 23 HRV features (Table 5.1) were extracted from the standard 5 min segments, whereas 6 ultra-short

HRV features (MeanNN, StdNN, MeanHR, StdHR, HF, and SD2) were extracted from 1 min HRV excerpts as they showed to be good surrogates of short HRV features in 1 min excerpts.



(a) Segmentation process for the SCWT. The last 5 min segments were extracted for both rest and stress conditions. The central 1 min segments were also extracted within the selected 5min segments for both rest and stress conditions.



(b) Segmentation process for the VGC. The last 5 min segments were extracted for both rest and stress conditions. The central 1 min segments were also extracted within the selected 5min segments for both rest and stress conditions.

Figure 5.20: Segmentation process for in-lab stress. 5min and 1min segments for both experiments during rest and stress were analysed.

The HRV analysis was performed using the Kubios software [32]. Time and frequency and non-linear features were analysed according to international guidelines in Chapter 2, section 2.4.1. Frequency domain features were extracted from power spectra estimated with autoregressive (AR) model methods.

5.3.5 Statistical analysis

A normality test was run to show that the HRV features are non-normally distributed. Therefore, the median (MD), standard deviation (SD), 25th and 75th percentiles were calculated to describe the distribution of HRV features over 5 min and 1 min. The non-parametric Wilcoxon's Signed-Rank Test was used to investigate the statistical significances between the stressor session and a baseline for both

experiments at 5 min and 1 min NN excerpts.

5.3.5.1 Comparison real VS in-lab stressors: exploratory analysis

In order to explore the effects of using real and in-lab stressors on short term HRV features (i.e., 5 minutes), a group analysis was performed as shown in Fig. 5.21. As for both experiments the same HRV features were reported, an interaction test was carried out. Short term HRV analysis was investigated as it is widely explored and assumed as the gold standard for HRV analysis.

After having investigated the effect of real and in-lab stressors within groups (i.e., rest VS stress), the effect size was investigated among groups using an independent, non-parametric statistical test: the Kruskal Wallis test. The null hypothesis was that no differences were observed across the studies. The cut-off value was a p-value less than 0.05.

The effect size was calculated as the absolute difference between rest and stress (i.e., ‘unstandardised’ difference) [284, 285]. The use of ‘unstandardised’ difference (i.e., raw difference between two groups) was preferable as the majority of the HRV features were significantly non-normally distributed [285]. In fact, when the outcome is reported on a meaningful scale and all studies in the analysis use the same scale, the effect size can be performed directly on the raw difference in values. The primary advantage of the raw difference is that it is intuitively meaningful.

Moreover, profile plots (median and standard error) of HRV features for all three experiments were reported for both rest and stress conditions.

However, it is important to take into account that the experiments were carried out using different stressors (i.e., an AE, a SCWT and a VGC), performed on different sample size and using different protocols, although the same inclusion and exclusion criteria were employed to enrol volunteers.

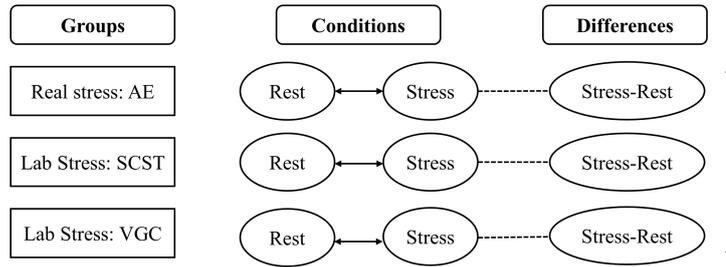


Figure 5.21: Conceptual difference between within- and between-group test. The horizontal arrow indicates a within-group test. The tests were performed using a Wilcoxon's Signed-Rank test and the results were p-values that measure the effect of the test within the group on HRV features. The vertical arrow indicates a between-group interaction test. This test was performed using a Kruskal-Wallis test and the results were p-values that measure the effect of the test between groups on HRV features.

5.3.6 Data-driven machine learning

A new classifier, different from that previously achieved using real-life data, was developed using ultra-short HRV features extracted from the two independent in-lab experiments. This was mainly due to the different stressors used (real VS lab stressors). Moreover, only ultra-short HRV features were used to develop the classifier in order to verify the relevance of using ultra-short HRV features to detect stress.

The ultra-short HRV features extracted from the SCWT were used to train and validate an automatic classifier to detect acute mental stress, as shown in Fig. 5.22. The ultra-short HRV features extracted from the VGC were used to test the model. The reason behind this choice was purely experimental in order to understand if the classifier was able to discriminate among different lab stressors. Moreover, although the two lab stressors showed different responses to mental stress in the behaviours of the HRV features, the ultra-short HRV features used to develop the classifier reported the same trends in both the SCWT and VGC.

The SCWT dataset was split into two folders: Folder 1 (40% of subjects, 52 subjects) was used for feature selection; Folder 2 (60% of subjects, 76 subjects) was used to train and validate the classifiers. The entire VGC dataset was used to test the model (Folder 3). The features selection process was two-staged: applying relevance and the redundancy analyses as described in section 5.2.7.1. Training and validation of the machine-learning models (including the model parameter tuning) were performed using a 10-fold person-independent cross-validation approach.

Five different machine-learning methods were used to train, validate and test the classifiers (SVM, MLP, IBK, C4.5 and LDA). Each of these methods was used with all the possible combinations of N out the D selected features (with D equal to the number of the selected features and N spanning from 3 to D). The best model was chosen as the classifier achieving the highest Area under the Curve (AUC), which is a reliable estimator of both sensitivity and specificity rates. The model was then tested on the VGC dataset (i.e., 42 subjects). Binary classification performance measures were adopted according to the standards reported in Chapter 2, section 2.4.2. In addition, a ROC curve for the best model was constructed.

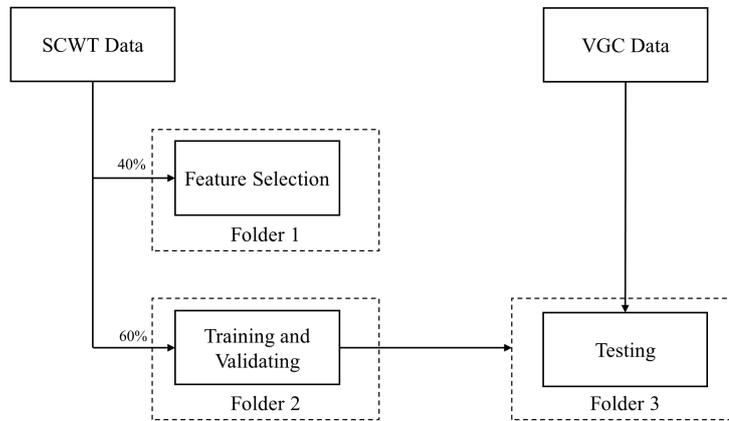


Figure 5.22: Data-driven machine learning workflow. The data acquired via SCWT were used to perform feature selection process, train and validate the classifiers. The data acquired via VGC were utilised as testing.

5.3.7 Results

128 healthy volunteers (49 females; age: 25 ± 3.85 years; BMI: 24.3 ± 4.7) with no history of heart disease, systemic hypertension or other diseases potentially influencing HRV were enrolled during the SCWT experiments. 42 healthy volunteers (12 females; age: 24 ± 4.5 years; BMI: 21.3 ± 1.5) with no history of heart disease, or other disease potentially influencing HRV were examined during the VGC experiments.

5.3.7.1 Statistical analysis

Normality test was run to show that HRV features are non-normally distributed. Therefore, the median (MD), standard deviation (SD), 25th and 75th percentiles and p-values were calculated for the HRV features extracted from 5 min and 1 min NN data series for SCWT and VGC experiments. Statistical analyses performed

are given in Tables 5.12, 5.13, 5.14 and 5.15 respectively.

Short term HRV analysis was investigated between rest and the SCWT (stress) conditions. As shown in 5.12, the MeanNN reported a significant decrease, whereas MeanHR and StdHR showed a significant increase over the stress session. During the SCWT session, LF and LF/HF ratio showed a significant increase while HF decreased. Most of the non-linear HRV features reported a significant decrease, while dfa1 reported a significant increase during the stress session. As also reported in Table 5.13, the HRV features computable in 1 min showed the same trends as the 5 min HRV features as already demonstrated in the previous study (section 5.2).

Table 5.12: HRV features in rest and stress from 5 min NN data series. SCWT experiment.

Short term 5 min										
HRV features	Rest				Stress				p-value	Trend
	MD	SD	25 th	75 th	MD	SD	25 th	75 th		
MeanNN (ms)	805.049	115.483	727.526	886.289	730.891	108.187	668.504	809.091	0.000	↓↓
StdNN (ms)	64.539	29.277	51.561	89.569	63.327	25.927	51.414	80.578	0.534	↓
MeanHR (1/min)	75.490	11.020	68.260	83.127	82.783	12.114	74.855	90.709	0.000	↑↑
StdHR (1/min)	6.286	3.102	5.098	8.187	6.974	5.395	5.851	9.099	0.029	↑↑
RMSSD (ms)	40.423	24.413	29.184	57.766	40.217	21.631	26.620	51.874	0.392	↓
NN50 (-)	64	53.965	26.5	117.5	66.5	48.485	27	99	0.832	↑
pNN50 (%)	17.157	17.008	7.010	33.525	17.705	13.391	5.994	26.301	0.353	↑
LF (ms ²)	1230.325	1574.108	736.124	2209.155	1696.763	1648.102	1028.244	2911.249	0.006	↑↑
HF (ms ²)	740.685	1220.06	346.590	1355.921	577.964	753.706	252.930	948.707	0.006	↓↓
LF/HF (-)	2.067	2.018	1.211	3.633	2.963	2.478	1.967	4.219	0.001	↑↑
TotPow (ms ²)	3830.58	4257.718	2431.51	6963.021	3727.483	3840.861	2404.829	5808.06	0.780	↓
SD1 (ms)	28.623	17.291	20.661	40.911	28.478	15.316	18.844	36.731	0.391	↓
SD2 (ms)	92.81	61.266	76.434	137.787	80.153	735.444	64.886	102.457	0.000	↓↓
ApEn (-)	1.078	0.109	1.006	1.138	1.094	0.105	1.014	1.156	0.366	↑
SampEn (-)	1.335	0.301	1.145	1.505	1.286	0.244	1.116	1.449	0.278	↓
D2 (-)	3.239	1.130	2.383	3.647	3.396	1.015	2.687	3.751	0.186	↑
dfa1 (-)	1.248	0.243	1.045	1.394	1.342	0.183	1.199	1.454	0.000	↑↑
dfa2 (-)	0.937	0.198	0.823	1.069	0.772	0.213	0.634	0.926	0.000	↓↓
RPlmean (beats)	11.831	6.948	9.360	14.496	9.840	3.709	8.568	11.966	0.000	↓↓
RPlmax (beats)	256	116.193	159	336	265.5	118.924	176.5	353.5	0.320	↑
REC (%)	35.769	11	30.107	42.313	31.142	8.211	26.584	37.158	0.001	↓↓
RPadet (%)	98.831	1.228	97.975	99.345	98.608	0.957	98.082	99.100	0.244	↓
ShanEn (-)	3.255	0.404	3.016	3.473	3.075	0.312	2.912	3.293	0.001	↓↓

MD: Median; SD: Standard Deviation; trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) under stress ($p > 0.05$). In bold HRV features changing significantly between rest and stress conditions.

Table 5.13: HRV features in rest and stress from 1 min NN data series. SCWT experiment.

Ultra-short term 1 min										
HRV Features	Rest				Stress				p-value	Trend
	MD	SD	25 th	75 th	MD	SD	25 th	75 th		
MeanNN (ms)	810.298	118.733	738.154	895.75	726.834	108.509	668.558	809.648	0.000	↓↓
StdNN (ms)	56.785	30.965	35.988	75.440	56.439	24.656	44.032	70.928	0.523	↓
MeanHR (1/min)	74.52	11.367	67.987	82.091	83.106	12.129	74.436	90.306	0.000	↑↑
StdHR (1/min)	5.067	3.33	3.717	7.181	6.416	3.625	4.991	8.007	0.000	↑↑
HF (ms ²)	689.102	1487.38	275.495	1232.26	415.356	661.036	208.708	819.132	0.001	↓↓
SD2 (ms)	96.129	49.324	70.550	145.008	73.109	32.076	56.553	92.867	0.000	↓↓

MD: Median; SD: Standard Deviation; trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) under stress ($p > 0.05$). In bold HRV features changing significantly between rest and stress conditions.

Short term HRV analysis was investigated between rest and the VGC (stress) conditions. As shown in Table 5.14, StdNN showed a significant decrease, while StdHR decreased significantly during the stress session. The frequency HRV features showed a significantly depressed trend over the VGC session, except for the LF/HF ratio. Most of the non-linear HRV features decreased during the stress session, except for ApEn and SampEn. Table 5.15 reports the HRV features computed over 1 min excerpts. All the HRV features reported the same trends in the equivalent 5 min HRV features as already demonstrated. This was a further demonstration that the ultra-short HRV features : MeanNN, StdNN, MeanHR, StdHR, HF and SD2, are good surrogates of short term HRV features.

Table 5.14: HRV features in rest and stress from 5 min NN data series. VGC experiment.

Short term 5 min										
HRV features	Rest				Stress				p-value	Trend
	MD	SD	25 th	75 th	MD	SD	25 th	75 th		
MeanNN (ms)	791.745	104.628	715.756	873.264	768.963	114.1736	677.085	850.922	0.359	↓
StdNN (ms)	65.430	29.584	53.170	92.028	48.146	23.856	36.820	63.105	0.000	↓↓
MeanHR (1/min)	76.768	10.629	69.737	85.441	78.783	12.797	70.910	89.019	0.436	↑
StdHR (1/min)	5.359	1.958	4.125	6.363	6.730	2.405	5.822	8.904	0.000	↑↑
RMSSD (ms)	39.428	21.372	28.571	56.426	32.342	20.852	23.138	46.358	0.111	↓
NN50 (-)	64	52.074	29	116.75	40	60.670	14	99.5	0.129	↓
pNN50 (%)	16.236	15.697	7.212	32.251	11.11	17.719	3.479	28.365	0.116	↓
LF (ms ²)	1439.429	1456.573	769.872	2272.208	711.467	859.242	373.944	1532.006	0.000	↓↓
HF (ms ²)	529.558	1024.198	325.887	938.927	267.596	837.695	174.1	588.753	0.004	↓↓
LF/HF (-)	2.3858	2.018	1.559	3.725	2.326	2.907	1.523	3.607	0.763	↓
TotPow (ms ²)	4201.23	4516.778	2629.507	7513.44	2044.663	2929.521	1219.117	3626.977	0.000	↓↓
SD1 (ms)	27.921	15.132	20.227	39.96	22.907	14.768	16.379	32.828	0.111	↓
SD2 (ms)	89.288	40.081	70.465	123.28	61.748	31.820	47.491	85.22	0.000	↓↓
ApEn (-)	1.082	0.0944	1.025	1.147	1.138	0.090	1.090	1.193	0.001	↑↑
SampEn (-)	1.280	0.246	1.122	1.462	1.499	0.254	1.327	1.651	0.000	↑↑
D2 (-)	3.341	1.112	2.365	3.741	2.768	1.35	1.189	3.751	0.170	↓
dfa1 (-)	1.270	0.204	1.158	1.406	1.166	0.247	1.007	1.309	0.006	↓↓
dfa2 (-)	1.003	0.162	0.899	1.117	0.933	0.185	0.807	1.077	0.027	↓↓
RPImean (beats)	12.482	6.659	10.285	15.929	10.554	4.717	8.65	14.317	0.012	↓↓
RPImax (beats)	284	95.879	224.75	358	259	143.054	105.25	360.25	0.065	↓
REC (%)	38.404	10.651	31.004	44.263	33.523	8.909	27.082	38.81	0.004	↓↓
RPadet (%)	99.033	0.965	98.365	99.457	98.369	1.436	97.303	99.02	0.001	↓↓
ShanEn (-)	3.326	0.383	3.140	3.557	3.148	0.353	2.921	3.468	0.016	↓↓

MD: Median; SD: Standard Deviation; trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress (p<0.05); ↓ (↑): lower (higher) under stress (p>0.05). In bold HRV features changing significantly between rest and stress conditions.

Table 5.15: HRV features in rest and stress from 1 min NN data series. VGC experiment.

Ultra- short term 1 min										
HRV Features	Rest				Stress				p-value	Trend
	MD	SD	25 th	75 th	MD	SD	25 th	75 th		
MeanNN (ms)	792.034	114.077	735.186	869.194	758.905	113.986	678.796	842.041	0.137	↓
StdNN (ms)	56.942	29.116	35.958	77.743	39.446	26.611	29.931	55.919	0.008	↓↓
MeanHR (1/min)	76.226	11.468	69.493	82.262	79.283	12.929	71.795	88.567	0.163	↑
StdHR (1/min)	4.058	2.426	3.461	5.633	5.3	2.587	3.710	7.253	0.042	↑↑
HF (ms ²)	411.148	850.439	248.185	889.212	246.649	919.256	165.290	595.443	0.029	↓↓
SD2 (ms)	72.136	40.079	46.540	105.426	48.776	35.743	38.796	73.214	0.004	↓↓

MD: Median; SD: Standard Deviation; trend analysis: ↓↓ (↑↑): significantly lower (higher) under stress (p<0.05); ↓ (↑): lower (higher) under stress (p>0.05). In bold HRV features changing significantly between rest and stress conditions.

5.3.7.2 Comparison real VS in-lab stressors: exploratory results

Exploratory results on the effect of real and in-lab stressors are presented in Tables 5.16 and 5.17. The results should be carefully assessed as the experiments were carried out on different samples, which could cause heterogeneity in the results.

Therefore, many differences among the experiments could be due to different sample size, the use of various stressors and protocols.

In Table 5.16 are reported the trends in the short HRV features during rest and stress sessions for the real and in-lab stressors. 18 out of 23 HRV features changed significantly between rest and stress using a real stressor (i.e., AE), whereas 12 and 14 out of 23 HRV features changed significantly between rest and the SCWT, and the VGC respectively. StdHR, LF, HF, SD2, dfa1, RPl_{mean}, REC and ShanEn changed significantly using real and in-lab stressors. In particular, StdHR (i.e., increased significantly under stress), HF and SD2 maintained the same trends (i.e., decreased significantly under stress) among the different stressors. This was a great result as StdHR, HF and SD2 were among the ultra-short HRV features that were assessed as good surrogates of short HRV features, and they maintained the same trends under different stressors.

In Table 5.17, median (MD), standard deviation (SD), 25th and 75th percentiles were calculated from the absolute difference between stress and rest for each short term HRV feature during an AE, a SCWT and a VGC. Moreover, the effects of the real and in-lab stressors were investigated for all of the features and the p-values are reported in Table 5.17. The effects were reported between an AE and a SCWT, an AE and a VGC, and a SCWT and a VGC. As reported in the last three columns of Table 5.17, most of the short HRV features changed significantly between the real and in-lab stressors. However, although there is no agreement (p-value<0.05) between most of the HRV features in real and in-lab stress, more analysis should be carried out using the same sample to confirm the hypothesis that a real stressor is more effective than an in-lab stressor. Furthermore, some differences were also reported between the two in-lab stressors, and a possible explanation could lie in the fact that they were carried out using different protocols.

The difference in effects and trends can also be observed through a visual inspection in Fig. 5.23 and 5.23a. Observing most of the HRV features it is evident that during the real stressor there was greater activation of the “fight or flight” response than during the in-lab stressors. In particular, MeanNN, StdNN, MeanHR, StdHR, LF, TotPow, ApEn, SampEn, D2, dfa1, Rpl_{mean}, Rpl_{max} and REC showed higher activation during real rather than in-lab stressors.

Table 5.16: HRV features' trends during real and in-lab stressors.

HRV Features	Trend in AE	Trend in SCWT	Trend in VGC
MeanNN (ms)	↓↓	↓↓	↓
StdNN (ms)	↓↓	↓	↓↓
MeanHR (1/min)	↑↑	↑↑	↑
StdHR (1/min)	↑↑	↑↑	↑↑
RMSSD (ms)	↑	↓	↓
NN50 (-)	↑	↑	↓
pNN50 (%)	↓	↑	↓
LF (ms ²)	↓↓	↑↑	↓↓
HF (ms ²)	↓↓	↓↓	↓↓
LF/HF (-)	↓↓	↑↑	↓
TotPow (ms ²)	↓↓	↓	↓↓
SD1 (ms)	↑	↓	↓
SD2 (ms)	↓↓	↓↓	↓↓
ApEn (-)	↓↓	↑	↑↑
SampEn (-)	↓↓	↓	↑↑
D2 (-)	↓↓	↑	↓
dfa1 (-)	↓↓	↑↑	↓↓
dfa2 (-)	↑	↓↓	↓↓
RPlmean (beats)	↑↑	↓↓	↓↓
RPlmax (beats)	↓↓	↑	↓
REC (%)	↑↑	↓↓	↓↓
RPadet (%)	↑↑	↓	↓↓
ShanEn (-)	↑↑	↓↓	↓↓

AE: Academic Examination; SCWT: Stroop Colour Word Test; VGC: Video Game Challenge; ↓↓
(↑↑): significantly lower (higher) under stress ($p < 0.05$); ↓ (↑): lower (higher) under stress
($p > 0.05$).

Table 5.17: Comparison between real and in-lab stressors.

	AE-Rest					SCWT-Rest					VGC-Rest					AE VS SCWT		AE VS VGC		SCWT VS VGC														
	MD	SD	25 th	75 th		MD	SD	25 th	75 th		MD	SD	25 th	75 th	p-val		MD	SD	25 th	75 th	p-val		MD	SD	25 th	75 th	p-val		MD	SD	25 th	75 th	p-val	
MeanNN (ms)	-227.641	93.980	-308.12	-183.124		-14.762	40.707	-45.251	8.631		-101.49	67.554	-162.979	-70.362	0.000		-101.49	67.554	-162.979	-70.362	0.000		-101.49	67.554	-162.979	-70.362	0.005		-101.49	67.554	-162.979	-70.362	0.005	
StdNN (ms)	-15.560	20.123	-29.745	-5.429		-6.670	19.062	-23.068	2.574		-8.124	17.177	-17.697	7.013	0.042		-8.124	17.177	-17.697	7.013	0.042		-8.124	17.177	-17.697	7.013	0.022		-8.124	17.177	-17.697	7.013	0.022	
MeanHR (l/min)	40.248	17.405	29.693	51.774		1.505	4.467	-0.918	4.671		10.829	7.679	7.012	18.024	0.000		10.829	7.679	7.012	18.024	0.000		10.829	7.679	7.012	18.024	0.000		10.829	7.679	7.012	18.024	0.000	
StdHR (l/min)	2.049	3.648	0.953	4.795		0.606	3.844	-0.494	1.975		1.285	2.616	-0.012	2.113	0.000		1.285	2.616	-0.012	2.113	0.000		1.285	2.616	-0.012	2.113	0.040		1.285	2.616	-0.012	2.113	0.040	
RMSSD (ms)	2.304	20.860	-12.421	16.345		-2.688	12.372	-8.180	3.265		-5.942	17.542	-17.862	2.285	0.629		-5.942	17.542	-17.862	2.285	0.629		-5.942	17.542	-17.862	2.285	0.057		-5.942	17.542	-17.862	2.285	0.057	
NN50 (-)	2.000	79.596	-39.000	66.000		6.000	30.939	-22.250	10.250		-9.000	39.247	-32.500	18.750	0.270		-9.000	39.247	-32.500	18.750	0.270		-9.000	39.247	-32.500	18.750	0.360		-9.000	39.247	-32.500	18.750	0.360	
pNN50 (%)	-1.647	14.705	-13.539	8.340		1.554	9.256	-8.009	11.660		-3.581	12.263	-14.136	1.586	0.831		-3.581	12.263	-14.136	1.586	0.831		-3.581	12.263	-14.136	1.586	0.220		-3.581	12.263	-14.136	1.586	0.220	
LF (ms ²)	-984.559	1176.168	-1799.0	-259.655		39.790	1050.25	-468.53	373.466		-221.61	1169.80	-712.515	140.856	0.050		-221.61	1169.80	-712.515	140.856	0.050		-221.61	1169.80	-712.515	140.856	0.051		-221.61	1169.80	-712.515	140.856	0.051	
HF (ms ²)	-127.903	489.226	-381.74	-25.477		-83.152	617.149	-253.07	74.581		-377.11	687.075	-691.768	-186.055	0.567		-377.11	687.075	-691.768	-186.055	0.567		-377.11	687.075	-691.768	-186.055	0.013		-377.11	687.075	-691.768	-186.055	0.013	
LF/HF (-)	-1.133	3.138	-2.781	1.115		0.283	2.104	-0.471	1.108		-1.084	3.691	-2.481	0.253	0.020		-1.084	3.691	-2.481	0.253	0.020		-1.084	3.691	-2.481	0.253	0.049		-1.084	3.691	-2.481	0.253	0.049	
TotPow (ms ²)	-1538.413	2353.215	-3228.4	-487.448		-688.41	2784.24	-2631.2	265.353		-865.63	2378.630	-1527.708	607.360	0.012		-865.63	2378.630	-1527.708	607.360	0.012		-865.63	2378.630	-1527.708	607.360	0.000		-865.63	2378.630	-1527.708	607.360	0.000	
SD1 (ms)	1.638	14.765	-8.798	11.552		-1.905	8.762	-5.798	2.313		-4.210	12.425	-12.648	1.613	0.631		-4.210	12.425	-12.648	1.613	0.631		-4.210	12.425	-12.648	1.613	0.049		-4.210	12.425	-12.648	1.613	0.049	
SD2 (ms)	-27.191	26.866	-49.277	-16.169		-12.382	29.948	-40.825	2.401		-28.528	28.181	-50.678	-16.690	0.034		-28.528	28.181	-50.678	-16.690	0.034		-28.528	28.181	-50.678	-16.690	0.834		-28.528	28.181	-50.678	-16.690	0.834	
ApEn (-)	-0.127	0.232	-0.252	0.045		0.021	0.117	-0.017	0.097		0.024	0.111	-0.073	0.083	0.000		0.024	0.111	-0.073	0.083	0.000		0.024	0.111	-0.073	0.083	0.026		0.024	0.111	-0.073	0.083	0.026	
SamPEn (-)	-0.338	0.388	-0.514	-0.106		-0.079	0.301	-0.105	0.299		0.121	0.284	-0.279	0.144	0.000		0.121	0.284	-0.279	0.144	0.000		0.121	0.284	-0.279	0.144	0.049		0.121	0.284	-0.279	0.144	0.049	
D2 (-)	-0.969	1.445	-2.738	-0.137		0.100	1.043	-0.361	0.594		-0.082	1.042	-0.674	0.418	0.000		-0.082	1.042	-0.674	0.418	0.000		-0.082	1.042	-0.674	0.418	0.000		-0.082	1.042	-0.674	0.418	0.000	
dfa1 (-)	-0.377	0.453	-0.712	0.043		0.013	0.226	-0.137	0.138		-0.251	0.267	-0.354	0.420	0.001		-0.251	0.267	-0.354	0.420	0.001		-0.251	0.267	-0.354	0.420	0.000		-0.251	0.267	-0.354	0.420	0.000	
dfa2 (-)	0.023	0.225	-0.183	0.100		-0.129	0.232	-0.293	0.101		-0.218	0.203	-0.367	-0.064	0.001		-0.218	0.203	-0.367	-0.064	0.001		-0.218	0.203	-0.367	-0.064	0.001		-0.218	0.203	-0.367	-0.064	0.001	
RPImean (beats)	3.073	7.681	-0.730	5.756		-1.928	7.797	-5.442	1.597		-1.070	6.692	-2.382	1.769	0.000		-1.070	6.692	-2.382	1.769	0.000		-1.070	6.692	-2.382	1.769	0.004		-1.070	6.692	-2.382	1.769	0.004	
RPImax (beats)	-93.500	168.496	-171.000	39.000		4.000	135.856	-125.75	69.250		-42.000	136.317	-115.250	18.750	0.138		-42.000	136.317	-115.250	18.750	0.138		-42.000	136.317	-115.250	18.750	0.000		-42.000	136.317	-115.250	18.750	0.000	
REC (%)	8.939	14.292	-1.264	17.282		-4.318	12.046	-12.473	3.637		-3.955	9.623	-6.752	5.141	0.000		-3.955	9.623	-6.752	5.141	0.000		-3.955	9.623	-6.752	5.141	0.002		-3.955	9.623	-6.752	5.141	0.002	
RPadet (%)	0.473	1.497	-0.732	0.837		-0.306	1.371	-1.096	0.378		-0.003	1.317	-0.467	0.693	0.021		-0.003	1.317	-0.467	0.693	0.021		-0.003	1.317	-0.467	0.693	0.008		-0.003	1.317	-0.467	0.693	0.008	
ShanEn (-)	0.298	0.491	-0.074	0.490		-0.204	0.481	-0.480	0.159		-0.097	0.386	-0.274	0.239	0.000		-0.097	0.386	-0.274	0.239	0.000		-0.097	0.386	-0.274	0.239	0.008		-0.097	0.386	-0.274	0.239	0.008	

MD: Median; SD: Standard Deviation, AE: Academic Examination, SCWT: Stroop Colour Word Test, VGC: Video Game Challenge

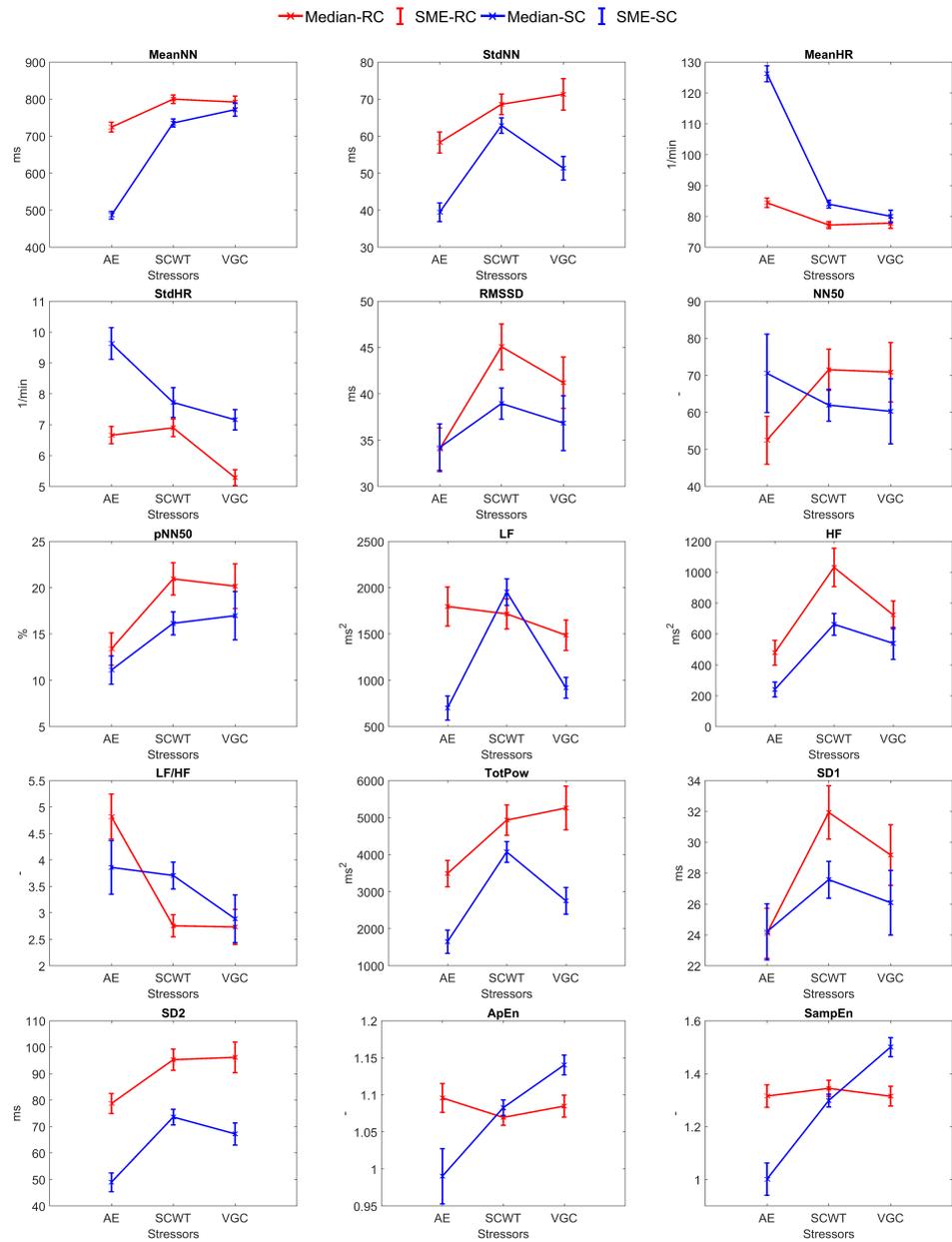


Figure 5.23: Profile plots of median and Standard Error (SEM) for the Rest Condition (RC) and the Stress Condition (SC) using different stressors. AE: Academic Examination; SCWT: Stroop Colour Word Test; VGC: Video Game Challenge; -:dimensionless.

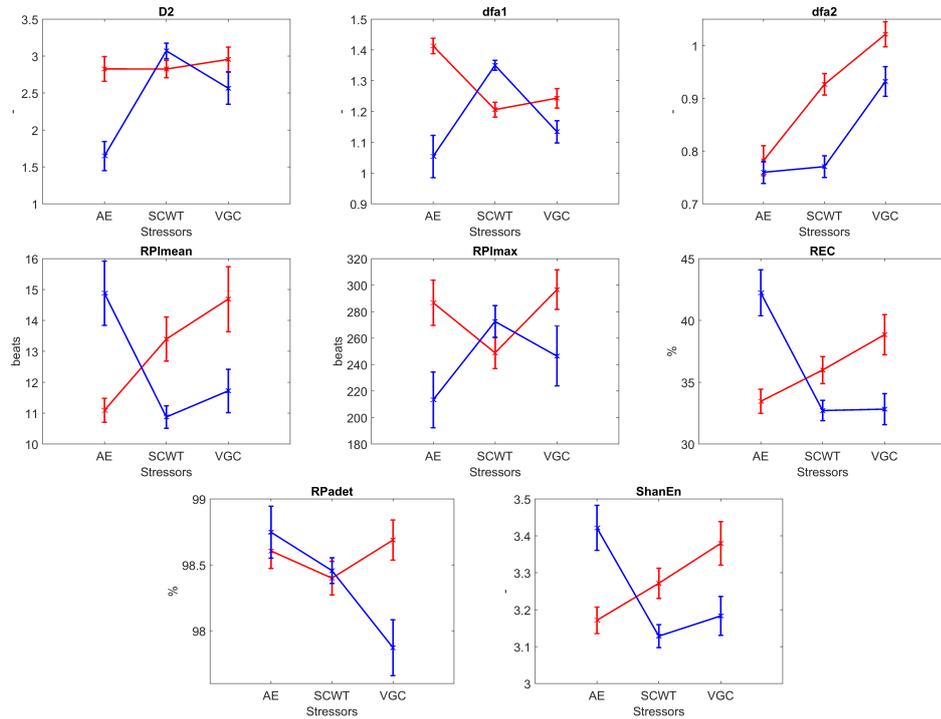


Figure 5.23a: Profile plots of median and Standard Error (SEM) for the Rest Condition (RC) and the Stress Condition (SC) using different stressors. AE: Academic Examination; SCWT: Stroop Colour Word Test; VGC: Video Game Challenge; -:dimensionless (cont.).

5.3.7.3 Classification and performance measurements

Regarding the feature selection, the 1-min HRV features (MeanNN, StdNN, MeanHR, StdHR, HF and SD2) were analysed in Folder 1. 5 (MeanNN, MeanHR, StdHR, HF and SD2) out of six HRV features also resulted in being relevant for this folder. This was not a trivial result given the lower number of subjects presented in Folder 1 than in the whole dataset. Among the 5 relevant features, 4 HRV features resulted non-correlated. However, MeanHR was not excluded as it was only highly correlated with MeanNN (Table 5.18). Consequently, all of the possible combinations of the 5 selected HRV features that proved relevant and non-redundant with each other (Table 5.19) were investigated.

Table 5.18: Correlation among HRV features in Folder 1.

	MeanNN	MeanHR	StdHR	HF	SD2
MeanNN	1	-0.980	-0.152	0.365	0.440
MeanHR		1	0.216	-0.318	-0.408
StdHR			1	0.265	0.424
HF				1	0.446
SD2					1

All the correlations resulted significant ($p_p < 0.05$); in bold Spearman's correlation coefficient (ρ) greater than 0.7.

Each machine learning method was trained and validated with all of the possible HRV feature combinations using Folder 2.

Table 5.19: Combinations of relevant and non-redundant HRV features.

ID	HRV Feature Combinations
1	MeanNN, StdHR, HF, SD2
2	MeanHR, StdHR, HF, SD2
3	MeanNN, StdHR, HF
4	MeanNN, StdHR, SD2
5	MeanNN, HF, SD2
6	MeanHR, StdHR, HF
7	MeanHR, StdHR, SD2
8	MeanHR, HF, SD2
9	StdHR, HF, SD2

The classifiers were then tested using Folder 3 (Table 5.20). According to the criteria defined in Chapter 4, the IBK classifier showed the highest AUC with 64% sensitivity, 85% specificity and 75% accuracy using MeanNN, StdHR, HF and SD2 as HRV features.

The performances of the IBK using different HRV feature combinations (Table 5.19) are shown in Fig. 5.24.

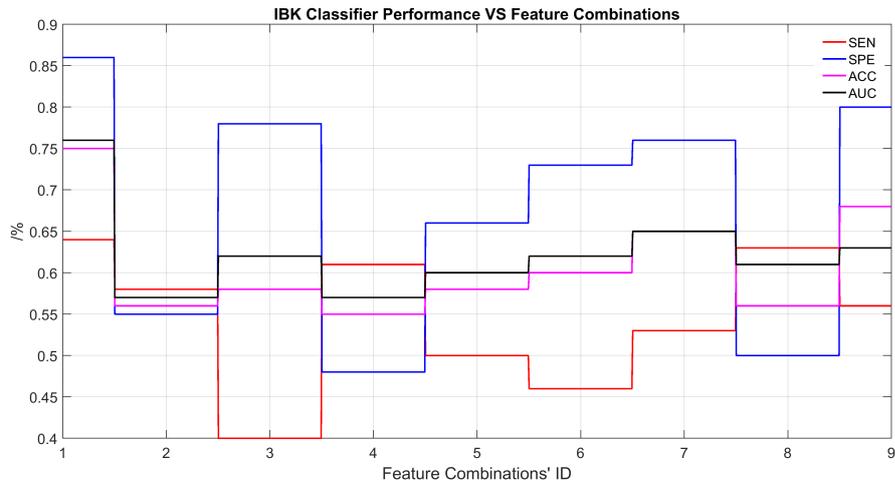


Figure 5.24: IBK performance against HRV features combinations.

Table 5.20: Model performance measurements estimated on test set (Folder 3) on 1 min excerpts.

Meth.	Parameters	AUC	SEN	SPE	ACC
MLP	LR=0.2; M=0.3; NE=800	62%	48%	79%	63%
SVM	RBF; G=8.5	63%	64%	62%	63%
C4.5	CF=0.0001; ML=2	63%	81%	57%	69%
IBK	K=1	75%	64%	86%	75%
LDA	-	57%	90%	29%	60%

Meth.: methods; MLP: Multilayer Perceptron; SVM: Support Vector Machine; C4.5: decision trees; IBK: Neighbor Search; LDA: Linear Discriminate Analysis; LR: Learning Rate; M: Momentum; NE= Number of Excerpts; RBF= Radial Basis Function kernel; G=Gamma; CF= Confidence Factor; ML= Minimum Number of Instances per Leaf; AUC: area under the curve; SEN: sensitivity; SPE: specificity; ACC: accuracy.

The ROC curve for the final model is shown in Fig. 5.25.

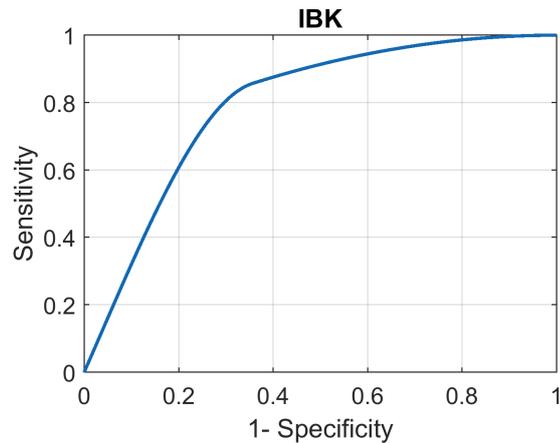


Figure 5.25: ROC curve of the IBK model developed using in-lab data via 1min HRV features.

5.3.8 Discussion

The current study aimed to investigate the validity of ultra-short HRV features to detect mental stress in a wider sample. Two different experiments were carried out in laboratory environments due to the degree of control and repeatability they provide. These experiments demonstrated that the subset of ultra-short HRV features (MeanNN, StdNN, MeanHR, StdHR, HF and SD2) selected in the previous study on real-life data were also able to discriminate between non-stressed and stressed subjects using in-lab stressors.

Regarding the results achieved using in-lab stressors, the statistical analysis computed on 5 min HRV features for the SCWT showed approximately the same results as for the VGC. However, some HRV features showed different behaviour between the SCWT and the VGC. In fact, LF and dfa1 decreased significantly over VGC whereas they showed a significant increase during the SCWT. This could be due to the different protocols employed in the two experiments. In fact, whilst during the SCWT the subject did not move, but only spoke, during the VGC the subject moved their hands to play, which, as already demonstrated during the meta-analysis in Chapter 3, section 3.2.2.13, could cause a change in some HRV features, especially for LF. Moreover, a significant increase in the LF power and a significant decrease in the LF/HF power showed a withdraw of the sympathetic function during the ‘fight or flight’ response in SCWT, which was less evident during the VGC. In the existing literature the majority of the studies have shown a significant increase of LF power, despite some studies having reported a decrease in LF during the stress session. However, those studies have used as a short-stressor, computer tasks and

physical tasks, which as stated in the previous chapters, elicit the SNS less than the other short-term stressors. The same explanation may be used for the different trends for LF/HF ratio, compared with the existing literature.

The HRV features from the SCWT and the VGC showed similar trends with the pooled HRV features (Table 3.6). In particular, MeanNN, StdNN, RMSSD, HF showed the same trends during the SCWT and the VGC, and with the pooled HRV features. Moreover, LF/HF showed a significant increase during SCWT in line with the pooled LF/HF (Table 3.6), whereas it showed a non-significant decrease during the VGC. Controversially, dfa1 showed a significant increase during the SCWT whereas it significantly decreased during the VGC in agreement with the pooled dfa1. One of reasons may be a higher activation of the sympathetic nervous system during the SCWT than VGC.

Other studies [116, 129, 140, 149, 159, 164, 286], employing the SCWT, also reported congruent results with this study, showing an increase in the LF and a decrease in vagal regulation of HRV (i.e., HF). As far as the non-linear HRV features are concerned, during the stress session, the majority of them showed a significant decrease in agreement with the literature. However, in contrast with the existing literature, dfa1 showed a significant increase.

The results from the VGC are also in agreement with the literature employing the same stressor [14, 122]. They reported an increase in HF and a decrease in the LF proving less parasympathetic activity during video games, as the LF component is now recognised to reflect both SNS and PNS [127, 287]. In fact, it can be speculated that LF could reflect the simultaneous dominance in sympathetic or vagal activation probably mediated through baroreflex, and these changes might depend on the individual intensity of different stressors to activate sympathovagal balance.

Indeed, some differences were shown among the real and in-lab stressors. The artificiality of the in-lab experiment settings may have produced unnatural behaviour that did not reflect real life, i.e., low ecological validity, and therefore, the results achieved using in-lab stressors may not be generalisable to a real-life setting. Although for many HRV features there was no agreement (a p-value less than 0.05) between real and in-lab stress, it cannot be claimed that they behaved completely differently. However, observing Figs. 5.23 and 5.23a, several HRV features showed higher activation during real than in-lab stressors. The differences reported in some HRV features' trends between real and in-lab stress could be due to stress-related changes in heart time irreversibility, which could depend on the underlying response system evoked by different stressors, i.e. active versus passive stress [288]. In this case, the active stress is something that subjects are actively in-

volved with (e.g., meeting a deadline of some type, academic examinations), and it is associated predominantly with cardiac beta-adrenergic (a class of sympathomimetic agents) activity, whereas passive stress (e.g., SCWT and VGC) elicits physiological responses reflecting alpha-adrenergic activation, which produces a lower response to stress than beta-adrenergic activity. Moreover, in active stressors, direct pathways from prefrontal regions are able to activate hypothalamic and brainstem autonomic control centres, leading to changes in the cardiac control system balance. In contrast, neurocardiac reactivity induced by a passive stressor is associated with autonomic regulatory subcortical centres (hypothalamus, brainstem), and at the level of the peripheral organ the heart. Therefore, diverse neurophysiological regulatory pathways could explain the controversial findings in heart rate time irreversibility indices between active (i.e., AE) and passive stressors (SCWT and VGC). However, some differences in HRV features between real and in lab-stressors could be also due to the different sample size and protocols employed in the various experiments.

Regarding the ultra-short HRV features (MeanNN, StdNN, MeanHR, StdHR, HF, and SD2) extracted from 1 min excerpts, they showed the same trends in both the SCWT and the VGC but also with the real stressor. Therefore, these results confirmed that the chosen subset of ultra-short HRV features was the one more reliable to detect stress. However, the model developed using real-life data was not compatible with in-lab data as the stressors used in the laboratory environments have been proved to be less stressful. Therefore, the six HRV features were also used to develop an automatic classifier to detect passive stress. The model used was IBK and it achieved sensitivity, specificity and accuracy rates of 64%, 86%, 75% respectively. The low efficacy of in-lab or passive stressors was also reflected in a lower sensitivity rate compared to the one achieved using real-life data. This model achieved a lower performance than previous studies [140, 148]. But the studies reported in Table 3.8 did not follow a robust methodology to develop the automatic classifier as many did not have a feature selection process or a testing procedure to validate the results.

5.3.9 Conclusion

This study proved that it is possible to detect stress using ultra-short HRV features, this result consolidated the previous results achieved in the study on real stress. In fact, the six ultra-short HRV features (MeanNN, StdNN, MeanHR, StdHR, HF, and SD2) also showed to be significant during in-lab stress and they maintained the same trends in both real and in-lab stress sessions.

However, the use of in-lab stressors proved to be less effective than a real

stressor. In fact, standard laboratory stressors do not always engage the subjects' affective response as social interaction stressors (e.g., public speaking tasks or academic examinations), which are often applied to provide a more appropriate social context in which negative emotions might be elicited. This was also shown by a drop in performance of the algorithm developed using in-lab data.

5.4 Conclusions and limitations

The continuing interest in everyday wearable devices being able to instantaneously assess mental stress levels is rising the attention in the scientific community around the use of HRV features computed in excerpts shorter than 5 min. Nevertheless, from the review of the existing literature, a gap was found in the investigation of mental stress using ultra-short HRV features. As a consequence, this study demonstrated that not all the ultra-short HRV features were good surrogates of short term ones. Only six ultra-short HRV features resulted to be good surrogates of short term ones: MeanNN, StdNN, MeanHR, StdHR, HF, and SD2. Those six features displayed consistency across all of the excerpt lengths (i.e., from 5 min to 1 min) and good performance if employed in a well dimensioned automatic classifier, which achieved sensitivity, specificity and accuracy of 82%, 94% and 88% respectively with only 1 min excerpts using a real-life stressor.

The efficacy of the ultra-short HRV features was also proved for a wider sample size using in-lab stressors (SCWT and VGC). In fact, the six ultra-short HRV features (MeanNN, StdNN, MeanHR, StdHR, HF, and SD2) also showed to be significant during in-lab stress and they maintained the same trends for both real and in-lab stress. However, the use of in-lab stressors showed to be less effective than real stressors. In fact, although the analysis performed was only exploratory, the effect of in-lab stressors showed to be lower than real stressor. Indeed, in-lab stressors elicit an engagement of the subject, which is different from real stressor eliciting a much stronger arousal. Therefore, the differences in HRV features among real and in-lab stressors could be mainly due to the elicitation of different mental states. The algorithm developed using in-lab data also showed a drop in performance achieving 64%, 85% and 75% for sensitivity, specificity and accuracy respectively, using 1 min excerpts.

In conclusion, although the results are promising, the effect of ultra-short HRV features was only investigated in healthy subjects with a limited age range (age between 20 and 40 years old) and using only real and cognitive stressors. Therefore, in order to generalise the presented results future experiments should investigate the

effect of ultra-short HRV features in a population with a wider age range. Moreover, breathing rate was not monitored during the acquisitions as it is usually monitored for respiratory sinus arrhythmia, even though it can cause quite significant changes in HRV features.

The next chapter presents the second case study and application of the framework presented in Chapter 4 to cope with unbalanced datasets.

Chapter 6

Cardiovascular and Autonomic Response to Falls in Later-life

6.1 Chapter overview

The previous chapter explored the CVS and ANS response to mental stress in real-life and in-lab settings. Pragmatic frameworks were applied: to investigate ultra-short HRV features as good surrogates of short ones, and to improve machine learning methods to cope with small datasets.

In this chapter, the framework presented in Chapter 4, section 4.2.2.3 to cope with unbalanced datasets via machine learning techniques is applied to a specific case study: fall prediction in later-life. In fact, in this chapter the relationship between the CVS and the ANS is investigated as a potential means to predict falls in later-life via HRV.

Fall prediction was chosen as case study not only because it is of relevant importance to the scientific community, but also because it is one of the best cases of rare events, which if, not predicted in time, could cause severe harm to the subject.

In particular, this chapter presents the results of a study aiming to develop a method to predict falls using short-term HRV analysis in hypertensive patients. This is a particular subgroup of older citizens because of drug prescriptions and prevalence of cardiovascular risk factors for falls for this group. Nevertheless, this is a significant subgroup, given the incidence of hypertension, which rises from the 60% in the 6th decade to the 70% in the 7th with a steep increase in the subsequent decades of life [289].

The study workflow is presented in Fig. 6.1. The main fall risk factors, prevention and prediction programmes along with the existing monitoring technologies

to detect and predict falls in the elderly were reviewed (deliverable 2a) in Chapter 3, section 3.3 and the need was highlighted to develop new classification methods based on non-invasive signals to predict rather than detect a fall. Therefore, since few studies have investigated HRV in fallers showing that there is a significant association between a depressed HRV and the risk of falling, a new study was developed to assess whether HRV could be used as a tool to predict falls (Fig. 6.1).

As opposed to the few previous studies investigating HRV in fallers [64, 224], which were focused on 24-hour HRV analysis, this is the first study describing the results obtained with short term HRV analysis, which is much easier and cheaper in terms of translation into everyday outpatient clinical practice. This study focused on short term HRV analysis and not on ultra short term HRV analysis as there is not a high demand for shortening HRV excerpts below the standard recommendations for fall prediction. Moreover, since this is the first study investigating HRV analysis below 24h for fall prediction, short term HRV analysis was preferred to ultra-short term analysis as short term HRV analysis is still considered the gold standard for HRV analysis.

The proposed approach is based on the idea that it is possible to detect constantly depressed ANS status early, which increases the risk of falling significantly. In fact, according to the existing literature, 42% of falls among the community-based older population are due to transient problems, which are significantly related to the CVS and the ANS conditions [290], including: gait/balance disorders, syncope, weakness, dizziness/vertigo, drop attacks and postural hypertension [174, 177, 178].

As opposed to other wearable technologies used in previous studies, HRV can be extracted from ECG, largely used to monitor and screen patients over 60 years old. In fact, ECG monitoring is beneficial for several cardiovascular diseases, and the application of ECG monitoring during real-life activities is under investigation for several purposes and particularly because of its effectiveness as early detector of cardiovascular diseases worsening [64, 291, 292]. Accordingly, most of the wearable and ambient sensing technologies aiming to monitor older subjects in real-life settings should include ECG or HRV monitoring.

Therefore, whilst older citizens could be sceptical of wearing technologies embedding accelerometers and gyroscopes “only” for falls prevention, it is expected that the same users would be less sceptical of adopting technologies that have been already proven to be effective for other cardiovascular diseases. In other words, enriching those technologies under exploration today with an ECG sensor could be a convenient combination in order to predict/detect a fall, while also being used to monitor cardiovascular problems. For these reasons, in this study, the popula-

tion considered was of hypertensive patients undergoing regular outpatient visits, for which ECG recordings were already be prescribed in order to monitor the risk of other cardiovascular events [64]. Moreover, other well-known risk factors for falls (e.g., multiple-prescriptions) are also systematically monitored and recorded in hypertensive patients undergoing regular outpatient visits, facilitating this study.

In this study, the most informative HRV features are investigated to predict falls in later life (deliverable 2b) and different from other methodologies used in previous studies, this study presents a model to automatically identify subjects at a higher risk of falling via HRV features (deliverable 2c) using advanced data mining methods for unbalanced datasets as described in Chapter 4, section 4.2.2.3. In fact, the methods used in this study reduced the number of false positive classification that are a considerable problem for many wearable devices.

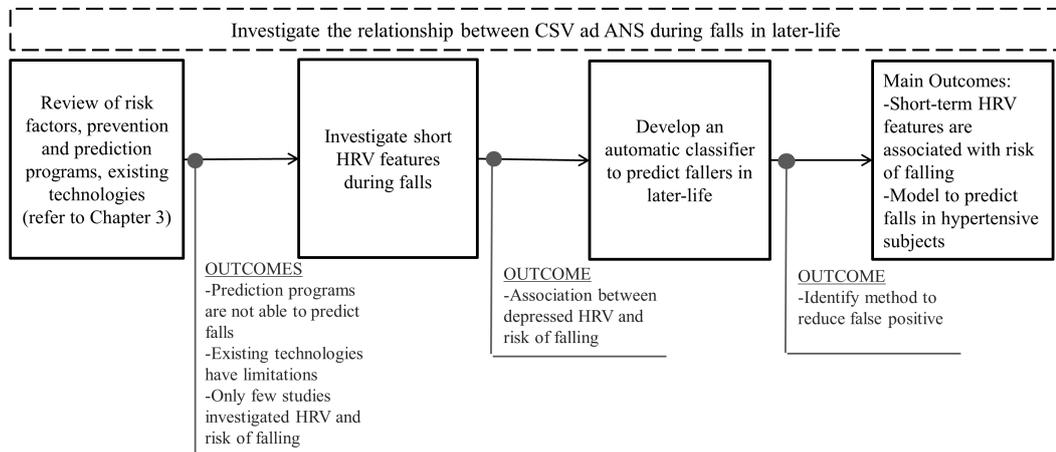


Figure 6.1: Workflow for Case Study 2. In order to investigate the relationship between the CVS and the ANS during falls, several steps have been undertaken to identify the short term HRV features that are associated with a risk of falling and develop an automatic classifier to predict falls in later-life.

In this chapter, objective 3 and individual deliverables referred to Case Study 2 are tackled.

6.2 Dataset

The data acquisition was carried out in the outpatient clinic for hypertension at the University Hospital of Naples Federico II, and therefore, it was approved by the Local Ethics Committee. All the participants signed specific informed consent to allow the use of their data for this study. The patients were recruited between the

1st January 2012 to the 10th November 2013 at the Centre of Hypertension of the University Hospital Federico II. Hypertensive patients were enrolled in this study if they met the following inclusion criteria:

- autonomous home dwelling over 55 years old;
- without cognitive impairments;
- without a history of falls in previous years.

For the baseline, a nominal 24h ECG Holter registration was performed, together with other periodic controls for hypertension management. The patients were hypertensive patients undergoing regular outpatient visits, for which ECG recordings were regularly prescribed in order to monitor the risk of other cardiovascular events. This avoided providing a further element of stress to the patient.

Demographic patient characteristics such as age, gender, BMI, Body Surface area (BMA), a history of stroke, diabetes and hypertension were recorded at the baseline along with blood pressure levels (diastolic and systolic), cholesterol values (Low (LDL) and High (HDL) Density Lipoprotein), history of hypertensive drugs (alpha- and beta-Blockers, ACE inhibitor and dihydropyridine) and finally, measurement of the intima-media thickness (IMT), left ventricular mass index and the blood ejection fraction to assess any history of cardiovascular problems.

The patients were followed up for 12 months after the recordings in order to record major cardiovascular and cerebrovascular events, i.e., fatal or non-fatal acute coronary syndrome including myocardial infarctions, syncopal events, coronary revascularization, fatal or non-fatal stroke, transient ischemic attacks and falls. All the events were adjudicated by the Committee for Event Adjudication in the Hypertension Centre. Adjudication was based on patient history, contact with the reference general practitioner and clinical records documenting the occurrence of the event/arrhythmia. However, this study was only focused on fall events. Falls were self-reported by patients. The following definitions for accidental falls were used in order to instruct patients and operators: “an unplanned descent to the floor with or without injuries” and/or “an event which results in a person coming to rest inadvertently on the ground or floor or some lower level” [175].

6.3 Hardware and software

Hardware ECGs were recorded using a Holter ECG Cardioscan DMS 300-3A and downloaded using the Cardioscan software (V12.0; DMS Holter, Stateside, NV, USA).

Software The different analyses were carried out using different software. The pre-processing of the ECG signals was carried out using the PhysioNet's toolkit as detailed in Chapter 5, section 5.2.2.

HRV analysis was carried out with Kubios. A full description of the software is also detailed in Chapter 5, section 5.2.2. All the statistical analyses were carried out using in-house tools developed in Matlab2016b.

Machine learning algorithms were developed using the Weka Platform (version 3.8.01) and Matlab2016b software.

6.4 Data analysis

The main stages of the data analysis, carried out for this study, are described in Fig. 6.2. The acquired ECGs were analysed and HRV features extracted from short excerpts (5 min). The statistical analyses were used to investigate the statistical significances of features' variation between fallers and non-fallers.

The methodology described in Chapter 4, section 4.2.2.3, was applied to this study in order to predict accidental falls in real-life settings via short term HRV features.

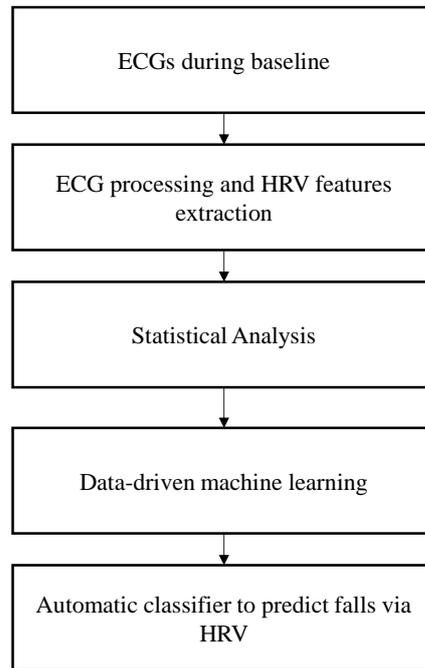


Figure 6.2: Data analysis flow. ECGs were acquired during a baseline assessment for hypertensive patients. The ECGs were pre-processed and HRV features extracted. Statistical analysis identified HRV features that changed significantly between fallers and non-fallers. Data-driven machine learning methods (NB, MNB, SVM, MLP, IBK) were used to develop an automatic classifier to predict falls via short term HRV features.

6.5 Short term HRV analysis

As shown in Fig. 6.3, the series RR beat intervals were obtained from ECG recordings using an automatic QRS detector based on a nonlinearly scaled ECG curve length feature [256]. The QRS detection was performed through the WQRS implementation, freely available from PhysioNet.

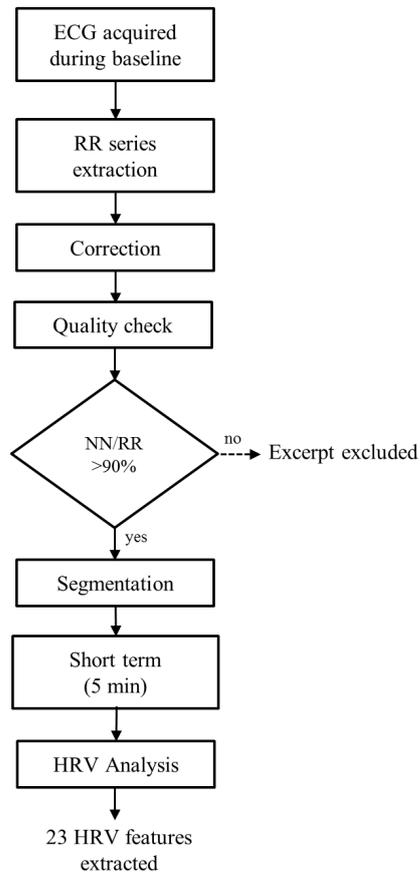
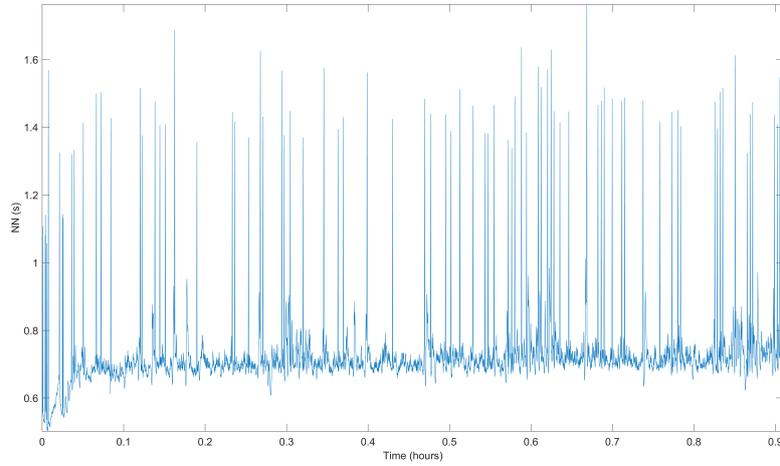
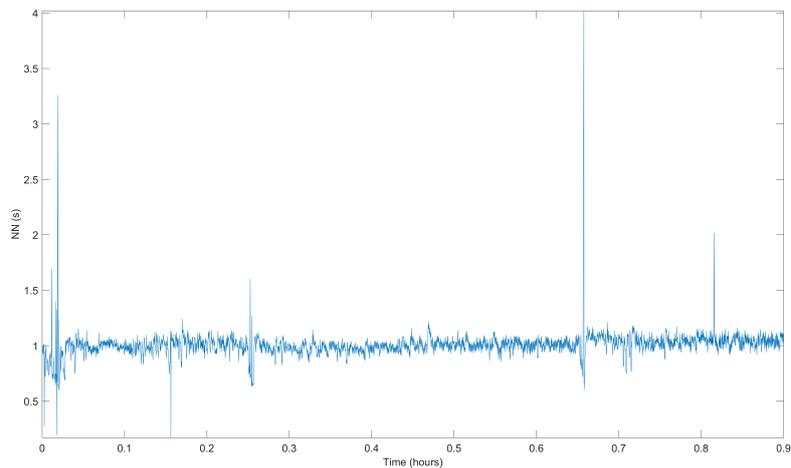


Figure 6.3: HRV processing workflow. NN/RR is the ratio of the total RR intervals labelled as NN (normal-to-normal beats). Short term HRV is analysed over 5 min excerpts.

An illustrative example of the raw NN (or RR, as no ectopic beats were detected) interval series for a faller and non-faller is shown in Fig. 6.4. However, no conclusions can be drawn from these raw NN series, therefore, short term HRV features were then extracted and analysed.



(a) Raw NN series for a faller.



(b) Raw NN series for a non-faller.

Figure 6.4: Raw NN series for one faller and non-faller during baseline session for over an hour.

All the Holter recordings were started in the early morning (i.e., from 8:30 am to 9:30 am). In order to avoid the “white coat effect” and maximally standardise the protocol (i.e., minimise heterogeneity due to the circadian cycle), the second and third hours of each recording were considered (approximately between 10:30 am and 12:30 pm). From these two hours the first 11 consecutive 5-minutes excerpts were used for the analysis. The two hours were initially selected and a quality check was performed using the NN/RR ratio. Each excerpt was only included among the consecutive 11 if the NN/RR ratio resulted in more than a value of 90%. Accord-

ing to the protocol, a subject would have been excluded if 11 consecutive excerpts would not have been identifiable in those two hours. This did not happen in the current study, since among the 11 segments no ectopic beats were detected or extracted. Standard linear HRV analysis according to International Guidelines [15] was performed. Moreover, non-linear features were computed according to the recent literature [45].

As shown in Table 6.1, time domain HRV features, reliable over 5 min HRV analysis, were calculated. The frequency domain HRV features were computed with AR methods. The generalised frequency bands in the case of the short term HRV recordings were low frequency (LF, 0.04-0.15 Hz) and high frequency (HF, 0.15-0.4 Hz). The included frequency domain features were absolute for each band, LF, HF, and the LF/HF power ratio. Non-linear HRV was analysed using the following methods: Poincaré Plot (PP), Approximate Entropy (ApEn), Correlation Dimension (CD), Detrended Fluctuation Analysis (DFA) and Recurrence Plot (RP).

Table 6.1: HRV features.

HRV Features	Units	Description
<u>Time Domain</u>		
MeanNN	[ms]	The mean of NN interval
StdNN	[ms]	Standard deviation of NN intervals
MeanHR	[1/min]	The mean heart rate
StdHR	[1/min]	Standard deviation of instantaneous heart rate values
RMSSD	[ms]	Square root of the mean squared differences between successive NN intervals
NN50	-	Number of successive NN interval pairs that differ more than 50 ms
pNN50	[%]	NN50 divided by the total number of NN intervals
<u>Frequency Domain</u>		
LF	[ms ²]	Low Frequency power (0.04-0.15Hz)
HF	[ms ²]	High Frequency power (0.15-0.4 Hz)
LF/HF	-	Ratio between LF and HF band powers
TotPow	[ms ²]	Total power
<u>Non Linear Domain</u>		
SD1, SD2	[ms]	The standard deviation of the Poincare' plot perpendicular to (SD1) and along (SD2) the line-of-identity
ApEn	-	Approximate entropy
SampEn	-	Sample entropy
D2	-	Correlation dimension
dfa1, dfa2	-	Detrended fluctuation analysis: Short term and Long term fluctuation slope
RPlmean	[beats]	Recurrence plot analysis: Mean line length
RPlmax	[beats]	Recurrence plot analysis: Maximum line length
REC	[%]	Recurrence rate
RPadet	[%]	Recurrence plot analysis: Determinism
ShanEn	-	Shannon entropy

6.6 Statistical analysis

A normality test was run to show that the HRV features are non-normally distributed. Therefore, the median (MD), standard deviation (SD), 25th and 75th percentiles were calculated to describe the distribution of the HRV features for fallers and non-fallers. The non-parametric Wilcoxon Signed-Rank test was used to investigate the statistical significances of the features' variation between fallers and non-fallers. The Wilcoxon Signed-Rank test was chosen as several HRV features, as expected, were not normally distributed. Baseline continuous and categorical variables were presented as the median (\pm standard deviation) or as count (percentage), respectively. The Wilcoxon Signed-Rank test and Chi-square test were adopted to compare continuous and categorical variables, respectively, between those who experienced a fall and those who did not.

6.7 Data-driven machine learning

According to the framework presented in Chapter 4, section 4.2.2.3, the whole dataset was split per subject into three folders (Fig. 6.5): Folder 1 (34%) was used for feature selection; Folder 2 (39%) was used for training and validating the classification models; finally, Folder 3 (27%) was adopted to evaluate the performance of the developed classification models. The subjects not included in Folder 1 were randomly assigned to Folder 2 or Folder 3 according to a 3:2 ratio. The reason for this asymmetric splitting was that Folder 2 was further split into 3 subsamples because of the 3-fold cross-validation technique.

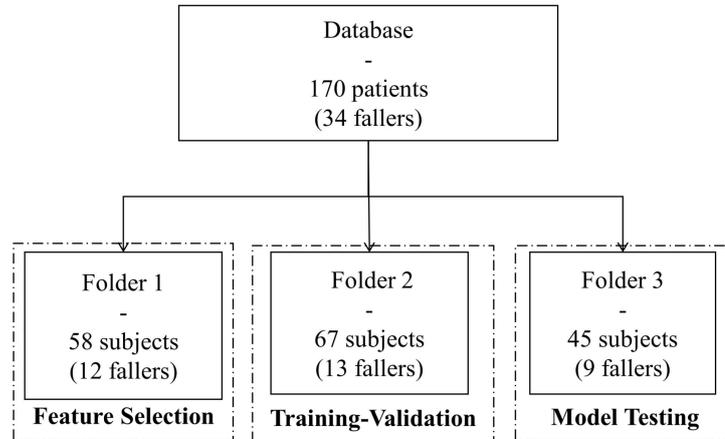


Figure 6.5: Splitting of the dataset into three folders. The whole dataset is split into three folders for feature selection, training and testing respectively.

6.7.1 HRV feature selection

Selecting the minimum set of features using the same folder utilised to train the machine learning model can reduce the generalisability of the final decisional model. Therefore, the HRV features were only minimised using Folder 1 (58 patients, of which there were 12 fallers). The feature selection was based on two main stages: the relevance analysis performed by the Wilcoxon Signed-Rank Test and redundancy analysis in terms of feature correlation as also described in Chapter 4, section 4.2.2.3.

The relevance analysis aimed to identify the HRV features changing more significantly among fallers and non-fallers. The Wilcoxon Signed-Rank test was adopted as not all the HRV features were normally distributed. All the HRV features changing significantly between fallers and non-fallers ($p\text{-value} < 0.05$) were selected at this stage. All of the relevant HRV features ($p\text{-value} < 0.05$) were then further minimised with the redundancy analysis aiming to exclude highly correlated features. Notions of measure redundancy were explored in terms of feature correlation via Spearman's rank correlation. The features with a Spearman's rank coefficient above 0.7 in absolute magnitude and with a significant $p\text{-value}$ (less than 0.05) were excluded. In this final stage, only the combinations of relevant and non-redundant HRV features were considered for the next steps. In this way, the feature selection process helped in the selection of a smaller set of significant features, simplifying the medical interpretation of the achieved results and directing attention only on the most important and informative parts of the signal.

6.7.2 Machine learning methods

Five different machine-learning methods were used to develop models aiming to automatically detect future fallers based on HRV features: Naïve Bayes (NB), which uses the Naïve Bayes' formula to calculate the probability of each class given the values of all of the attributes and assuming conditional independence and Gaussian distribution of the attributes; Multinomial Naïve Bayes (MNB), which is based on Bayes' theorem (Bayes rule), with the additional incorporation of frequency information and a multinomial distribution for each of the features; a Support Vector Machine (SVM), which belongs to a general field of kernel-based machine learning methods used to efficiently classify both linearly and non-linearly separable data; a Multilayer Perceptron (MLP) consisting of an artificial neural network of nodes (processing elements) arranged in layers; a K-Nearest Neighbour Classifier (IBK), which finds a group of K object in the training set that are closest to the test object, and bases the assignment of a label on the predominance of a particular class in this neighbourhood. Regarding the model parameters: for the MLP classifier, the learning rate (LR) varied from 0.3 to 0.9, the momentum (M) from 0.2 to 1 and the number of epochs from 100 to 2000 [259]; for the SVM, basis function kernel was used varying gamma (G) from 10^5 to 10 [260]; for the IBK, K varied from 1 to 5 [262]. The model parameters were tuned during training in Folder 1. The best parameters for each method were chosen as the ones that optimise their overall accuracy.

Each of those methods was used with all the possible combinations of N out the D selected features (with D equal to the number of features selected and N equal to 3 as the subjects presenting the event to predict (fallers) were 34 in number.).

6.7.3 Training, validation and testing

The training of the machine-learning models was performed on Folder 2 (67 patients, of whom 13 patients experienced a fall). Folder 2 was further divided into 3 equal sized subsamples, according to the 3-fold person-independent cross-validation approach. Of these 3 subsamples, 2 subsamples were used as training data and the remaining one was retained for validating the model. The process was then repeated 3 times, with each of the 3 subsamples used exactly once as the validation data. Finally, the cross-validated estimations were computed by averaging the performances over the 3 validation subsamples. Binary classification measures were adopted according to the standard formulae reported in Chapter 2, section 2.4.2, Table 2.5. Nevertheless, given the relatively small and unbalanced number of events

(falls) in each subsample; the random allocation of one subject to one of the three subsamples can significantly alter the cross-validation estimates. Therefore, the cross-validation procedure was repeated 10 times and the cross-validation estimates were averaged over those 10 iterations. This procedure was performed 5 times: one for each machine-learning method used to develop the predictive models (Fig. 6.6).

Testing a classifier involves analysing its performances on a set of subjects that is independent from the training and validation set. Accordingly, Folder 3 (45 patients) was used to test the trained models. Finally, the best performing model was selected as the one achieving the highest averaged AUC, which is a reliable estimator of both sensitivity and specificity rates and, in the case of an equal AUC average, the model with minimal structural complexity. In this case, model structural complexity refers to the number of features included in the predictive model [74, 76].

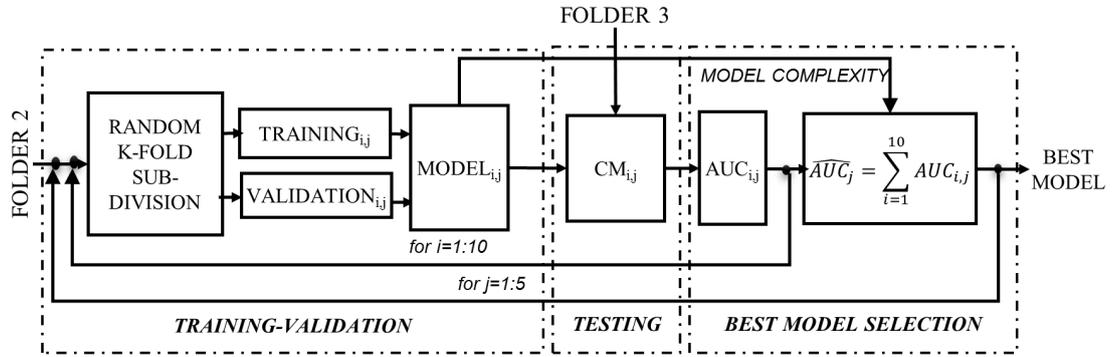


Figure 6.6: Model training, validation and testing. For each of the 5 learning-machine methods used ($j=1, \dots, 5$), the training-validation procedure was repeated 10 times ($i=1, \dots, 10$). For each iteration, the Confusion Matrix ($CM_{i,j}$) and the $AUC_{i,j}$ were calculated. The best method was the one with the max \overline{AUC}_j .

6.7.3.1 Final model generation

For the best performing method, a meta-model was produced by averaging the coefficients of the hyperplanes separating fallers from non-fallers for each of the 10 models generated during the validation process. The performances of this final model were computed using Folder 3. In addition, the Diagnostic Odds Ratio (DOR) was computed and the ROC curve for the best model was constructed.

6.8 Results

The current study was performed enrolling 170 hypertensive patients (including 56 females and 114 males), age 72 ± 8 years, of which 34 subjects experienced a fall within 3 months from the baseline assessment. The patients' characteristics are reported in Table 6.2. According to the baseline data, no statistically significant differences were observed between fallers and non-fallers. In other words, baseline characteristics were not able to distinguish between fallers and non-fallers.

Table 6.2: Patient baseline characteristics.

Clinical features	Non-Fallers MD\pmSD	Fallers MD\pmSD	p-value
Age (Years)	71.85(\pm 7.046)	70.33(\pm 9.6)	0.22
Gender (Female)	45(26.7%)	12(7.14%)	0.93
History of Hypertension	46(27.8%)	12(7.2%)	0.90
History of stroke	13(7.8%)	2(1.2%)	0.43
Diabetes	22(13.1%)	5(3%)	0.68
Diastolic BP(mmHg)	76.00(\pm 8.97)	75.92(\pm 11.75)	0.69
Systolic BP (mmHg)	136.76(\pm 20.15)	144.44(\pm 21.31)	0.06
Total cholesterol	176.96(\pm 36.34)	188.86(\pm 40.99)	0.13
LDL(mg/dl)	101.56(\pm 30.012)	113.67(\pm 35.16)	0.11
HDL(mg/dl)	52.25(\pm 13.33)	51.33(\pm 13.61)	1.00
BMI(kg/m²)	27.76(\pm 4.06)	27.27(\pm 4.13)	0.43
BSA(m²)	1.89(\pm 0.16)	1.9(\pm 0.22)	0.84
Alpha-blockers	21(12.6%)	7(4.2%)	0.64
Beta-blockers	56(33.7%)	13(7.8%)	0.45
ACE inhibitor	45(27.1%)	14(8.43%)	0.64
Dihydropyridine	35(21.08%)	9(5.4%)	0.82
IMT Mean(mm)	1.57(\pm 0.45)	1.41(\pm 0.36)	0.07
IMT Max(mm)	2.35(\pm 0.75)	2.23(\pm 0.89)	0.19
LVMi(g/m²)	131.84(\pm 26.32)	133.62(\pm 22.99)	0.68
EF(%)	58.90(\pm 11.24)	63.47(\pm 6.51)	0.05

BP: blood pressure; BMI: Body Mass Index; BSA: Body Surface area; IMT: intima-media thickness; LVMi: left ventricular mass index; EF: ejection fraction.

6.8.1 Statistical analysis

Table 6.3 reports the median (MD), the standard deviation (SD), the 25th and the 75th percentiles for the 23 HRV features extracted from faller and non-faller patients for the whole dataset. The last column of Table 6.3 shows the Wilcoxon Signed-Rank test p-values for the features' variation between fallers and non-fallers. As shown in Table 6.3, 21 out of 23 HRV features changed significantly between fallers and non-fallers. In particular, lower values for all of the time domain features except

MeanHR was observed in fallers. Moreover, LF, HF and total power were lower in fallers, while LF/HF increased. This result indicated a sympathetic dominance or parasympathetic withdrawal in fallers. Furthermore, the statistical analysis showed that fallers had significantly lower SD1, SD2 and D2, which are positively correlated with parasympathetic withdrawal. The statistical analysis also showed significantly higher ApEn, SampEn, dfa1, dfa2, D2, REC, RPl_{max}, RPl_{mean}, RPadet, and ShanEn values. These results indicated low predictability of fluctuations in successive NN intervals in fallers. In other words, the cardiovascular system becomes less responsive to internal or external stimuli in fallers than non-fallers.

The statistical analysis was also repeated for each of the eleven 5-min segments and neither the p-values or the features' trends changed from the analysis reported in Table 6.3. This consolidated the idea that short term HRV analysis could be used to identify the risk of falling.

Table 6.3: HRV features in non-fallers and fallers.

HRV Features	Non-Fallers				Fallers				p-value	Trend
	MD	SD	25 th	75 th	MD	SD	25 th	75 th		
MeanNN (ms)	773.751	244.921	640.612	899.311	782.92	185.511	676.532	901.701	0.162	↑
StdNN (ms)	57.351	64.911	35.600	91.913	46.212	70.110	30.800	73.810	0.000	↓↓
MeanHR (1/min)	83.862	10.224	77.114	92.445	124.744	17.845	111.200	132.651	0.001	↑↑
StdHR (1/min)	5.753	1.991	4.241	6.458	3.923	1.8923	4.623	5.221	0.185	↓
RMSSD (ms)	48.655	64.721	26.254	86.456	29.255	83.489	19.901	50.400	0.002	↓↓
NN50 (-)	30.000	39.911	11.000	62.000	16.256	35.545	6.000	28.000	0.000	↓↓
pNN50 (%)	9.425	16.552	3.500	22.656	4.855	12.785	1.600	9.100	0.003	↓↓
LF (ms ²)	1000.201	950.232	100.110	2000.000	700.123	900.111	500.000	2000.001	0.001	↓↓
HF (ms ²)	1600.000	900.010	200.201	800.000	600.021	202.036	400.236	3400.000	0.000	↓↓
LF/HF (-)	6.625	1.537	0.466	11.085	9.712	2.014	0.575	21.123	0.001	↑↑
TotPow (ms ²)	2500.236	2143.254	1465.220	4478.002	735.001	2035.365	282.002	1335.365	0.005	↓↓
SD1 (ms)	34.448	45.910	18.587	61.222	20.711	59.151	14.112	35.612	0.000	↓↓
SD2 (ms)	71.726	79.720	43.260	115.552	60.597	72.626	40.501	91.432	0.003	↓↓
ApEn (-)	0.944	0.216	0.765	1.052	0.960	0.235	0.771	1.071	0.001	↑↑
SampEn (-)	1.060	0.516	0.701	1.453	1.238	0.577	0.752	1.612	0.000	↑↑
D2 (-)	0.902	0.287	0.714	1.106	1.035	0.389	0.783	1.261	0.001	↑↑
dfa1 (-)	0.876	0.295	0.666	1.078	0.975	0.323	0.753	1.151	0.000	↑↑
dfa2 (-)	0.800	1.390	0.065	2.377	0.464	1.346	0.048	1.903	0.005	↓↓
RPlmean (beats)	0.445	0.161	0.325	0.525	0.451	0.156	0.367	0.534	0.004	↑↑
RPlmax (beats)	126.001	106.456	67.002	212.002	184.002	109.900	111.001	289.236	0.000	↑↑
REC (%)	15.275	14.790	9.735	23.044	16.861	14.901	11.812	24.910	0.000	↑↑
RPadet (%)	0.990	0.022	0.981	1.000	0.990	0.010	0.991	1.000	0.001	↑↑
ShanEn (-)	3.344	0.589	2.970	3.781	3.421	0.581	3.211	3.888	0.001	↑↑

MD.: Median; SD: Standard Deviation; Trend; ↓↓ (↑↑): significantly lower (higher) in fallers ($p < 0.05$); ↓ (↑): lower (higher) in fallers ($p > 0.05$). In bold HRV features changing significantly between fallers and non-fallers.

6.8.2 Classification and performance measurements

HRV feature selection, performed on Folder 1, showed that 16 HRV features were relevant (i.e., changed significantly also in this folder). Among the 16 relevant

features, the correlation matrix was produced as shown in Table 6.4. Consequently, all the possible 3-feature combinations of those 16 features that resulted as relevant and non-redundant with each other were investigated.

Table 6.4: Correlation among HRV features in Folder 1. All the correlation resulted significant ($p_\rho < 0.05$); in bold Spearman's correlation coefficient (ρ) greater than 0.7.

	RMSSD	NN50	pNN50	LF	HF	TotPow	SD1	ApEn	SampEn	dfa1	D2	REC	RPImax	Rplmean	Rpadet	ShanEn
RMSSD	1	0.447	0.588	0.444	0.467	0.486	1.000	-0.377	-0.332	-0.396	0.061	0.041	-0.350	-0.072	-0.070	-0.185
NN50		1	0.916	0.420	0.469	0.448	0.447	0.049	-0.032	-0.344	-0.068	-0.342	-0.486	-0.277	-0.398	-0.431
pNN50			1	0.426	0.473	0.453	0.588	-0.031	-0.020	-0.327	-0.140	-0.383	-0.520	-0.293	-0.478	-0.484
LF				1	0.744	0.893	0.444	-0.024	-0.089	-0.569	-0.024	-0.008	-0.418	-0.247	-0.152	-0.339
HF					1	0.940	0.467	0.008	-0.063	-0.614	-0.052	-0.002	-0.539	-0.219	-0.286	-0.368
TotPow						1	0.486	-0.039	-0.098	-0.599	-0.019	0.043	-0.492	-0.221	-0.194	-0.336
SD1							1	-0.377	-0.332	-0.396	0.060	0.041	-0.350	-0.072	-0.070	-0.185
ApEn								1	0.616	-0.186	-0.438	-0.260	-0.102	-0.461	-0.295	-0.335
SampEn									1	-0.112	-0.634	-0.422	-0.169	-0.460	-0.492	-0.422
dfa1										1	0.186	0.154	0.517	0.403	0.356	0.515
D2											1	0.523	0.339	0.517	0.522	0.522
REC												1	0.438	0.652	0.782	0.772
RPImax													1	0.578	0.552	0.683
Rplmean														1	0.535	0.849
Rpadet															1	0.785
ShanEn																1

Table 6.5 reports the performance measurements (mean and standard deviation) estimated on the independent test set for the 5 models, averaged over the 10 iterations. According to the criteria, the Multinomial Naïve Bayes' model outperformed the other data-mining methods achieving the best mean value of performance measures over 10 iterations: 72% sensitivity, 61% specificity and 68% accuracy.

Table 6.5: Performance measurements (Mean± SD) estimated on the test set (Folder 3).

Method	Parameters	AUC (%)	SEN (%)	SPE (%)	ACC (%)
NB	-	68±7.5	55.6±24.7	75.0±13.2	72.2±8.7
MNB	-	70.0±7.8	72.2±10.9	61.1±7.4	67.8±5.9
SVM	RBF; G=1.6	58.0±10.2	22.2±11.6	81.9±9.9	68.9±7.8
MLP	LR=0.6; M=0.4; NE=1800	60.0±6.2	5.6±16.5	84.7±9.3	71.1±6.0
IBK	K=1	54.0±10.3	22.2±12.2	79.2±6.4	67.8±6.1

NB: Naïve Bayesian; MNB: Multinomial Naïve Bayesian; SVM: Support Vector Machine; MLP: Multilayer Perceptron; IBK: Neighbor Search; RBF= Radial Basis Function kernel; LR: Learning Rate; M: Momentum; NE= Number of Excerpts; AUC: area under the curve; SEN: sensitivity; SPE: specificity; ACC: accuracy.

A meta-model was generated by averaging the coefficients of the hyperplanes separating fallers and non-fallers obtained at each iteration of the training and validation process. In log-space, the Multinomial Naïve Bayes' meta-model equation was:

$$-0.20SD1 + 0.05RPl_{max} - 0.05ShanEn - 0.59 \cong 0 \quad (6.1)$$

The interpretation of equation 6.1 could be the following: “a subject is classified as a faller they lie above the hyperplane”. In other words, a subject is identified at high risk of falling if:

$$-0.20SD1 + 0.05RPl_{max} - 0.05ShanEn - 0.59 > 0 \quad (6.2)$$

Using this interpretation, the Diagnostic Odds Ratio (DOR) for this model was 4.9 (CI 95%: 1.49 - 11.7). The ROC curve for the model estimated from the independent test set is shown in Fig. 6.7.

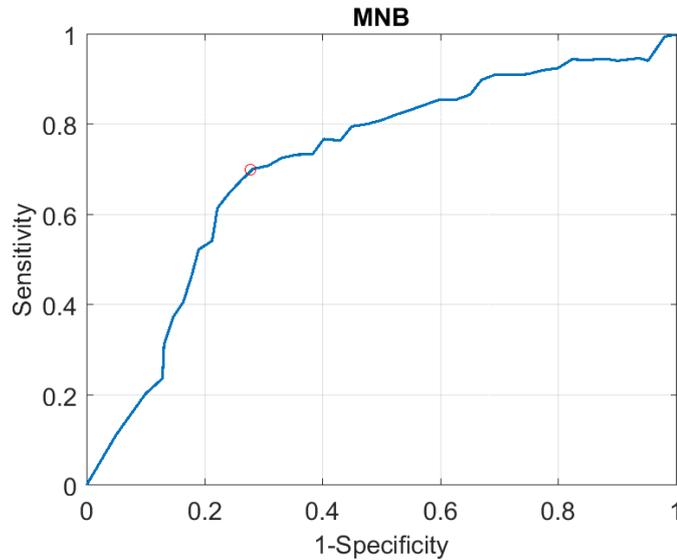


Figure 6.7: ROC curve of the Multinomial Naïve Bayes' final model.

6.9 Discussion

The current study proposed a mathematical model to automatically assess the risk of first-time falls in hypertensive patients, based on a few HRV features (SD1, RPI_{\max} and ShanEn). These features were extracted from 11 consecutive 5-minutes HRV excerpts extrapolated from the second and third hour of Holter registrations (approximately between 10:30 am and 12.30 pm).

The statistical analysis showed that fallers presented generally depressed time and frequency HRV features and an increase in non-linear heartbeat dynamics. It is known that HRV depression can be due to drug therapy or ageing. However, the results suggested that this difference was not due to those factors, because, as reported in Table 6.2, no statistically significant differences were observed in drug therapy or age between the fallers and non-fallers.

These results confirmed the previous findings on long term HRV analysis [64]. Also, another study [224] investigating HRV features' changes between fallers and non-fallers demonstrated the same features' trends for MeanNN, StdNN, pNN50 and LF, although they did not find statistically significant differences between fallers and non-fallers. This result could be due to several reasons, including the following: in [224] only linear long term HRV analysis was performed; it enrolled a smaller sample size (about 60 patients); they used the history of falls and not future falls to classify the subjects.

In this study, statistically significant differences in both linear and non-linear HRV features emerged between fallers and non-fallers. However, the non-linear ones appeared to have better discrimination ability: the 3 non-linear features selected during the feature selection phase (SD1, RPI_{max}, ShanEN) were then utilised independently by each machine learning method. During the testing, the best performances were achieved by the Multinomial Naïve Bayes' model, with relatively high sensitivity (72%), specificity (61%), accuracy (68%) and a DOR of 4.9 (CI 95% 1.49-11.7). The other models (NB, SVM, MLP, IBK) achieved high specificity and accuracy but quite low sensitivity during the testing. One of the reasons might be that these methods were not able to deal with the nature of the dataset (i.e., highly unbalanced), whereas Multinomial Naïve Bayes' method showed better performance in dealing with an unbalanced dataset. Therefore, Multinomial Naïve Bayes model was selected as the best model according to the highest AUC. However, for other applications (e.g., screening), other criteria (e.g., the highest sensitivity rate) could represent a better choice.

In a previous study [64], an automatic classifier based on 24-hours HRV features was proposed achieving a DOR of 4.2 (CI 95% 2.0-8.7). As opposed to Melillo *et al.* [64], the current study achieved better results by using 1-hour recordings and analysing the HRV on 5 min excerpts (short term analysis). Furthermore, the model presented in the current study was developed through a rigorous training, validation and testing procedure, using three independent subsets of data for feature selection, model training and testing and averaging the performances by repeating the procedure ten times. Moreover, this method based on HRV showed higher DOR, sensitivity and specificity to predict falls than functional mobility tests (Table 3.10) [195, 196].

The results presented in this study reinforce the idea that dysfunctions between the CVS and the ANS are associated with a higher risk of falling and can be used to predict future fallers. According to previous findings [64], the reasons could be that a depressed HRV reflects a reduced capability to react to extrinsic risk factors avoiding falls.

6.10 Conclusions, applications and limitations

The current study proposed a method based on short term HRV analysis to automatically identify future fallers among hypertensive patients aged 55 or over. The presented classifier achieved satisfactory results through a rigorous validation procedure, enabling to predict fallers with a sensitivity rate of 72% and a specificity

rate of 61%. Moreover, the method used to develop the machine learning model, described in Chapter 4, section 4.2.2.3, achieved better results than the study conducted on the same dataset by Melillo *et al.* [64], which used standard algorithms for unbalanced datasets without splitting the dataset into folders and only using a cross-validation procedure without testing.

The method proposed in this study is clinically feasible since it only requires 1-hour ECG recordings, which are often performed in cardiovascular patients also through wearable devices [293]. For instance, the method proposed does not require the use of other technologies such as wearable accelerometers or pressure matrices, which are not used in everyday clinical practice. For this reason, the method proposed could be easily integrated along with other clinical tools for estimating the risk of falling and could be widely used in outpatient settings to identify high-risk patients who need further assessment and could benefit from fall prevention programmes or fall detection systems [294–298]. This is important as falls depend on hundreds of risk factors and the integration of complimentary approaches could be more effective in predicting falls. For instance, the mechanisms that accelerometers or gyroscopes use could perform better across a population in which other intrinsic risk factors are frequent. In fact, focusing on hypertensive patients may have narrowed the study to a population where risk factors for falls due to cardiovascular problems were prevalent. Therefore, although hypertension affects 60% of people in the 6th decade of life, 70% in the 7th and so far, future studies on a different population and combining different approaches seems to be needed.

In fact, this study presents few limitations that should be considered before adopting this method in other contexts. This study focused on hypertensive patients, which represent special a population with distinguished characteristics, different from the population of community-dwelling older citizens. The patients were enrolled in an outpatient clinic for hypertension and not in a falls clinic. Therefore, important information, such as the exposure to other independent intrinsic/extrinsic risk factors for falls could not be assessed or used to independently verify the results. Moreover, the falls recordings were based on patient self-reports and potentially relevant characteristics of the recorded falls were not systematically recorded. Although patients were instructed on how to report a fall [175], the subjective quality of the data could have biased the results. Therefore, future work should consider this variable more carefully.

The next chapter presents the main conclusions of the thesis and how limitations could be addressed in future work.

Chapter 7

Conclusions and Future Work

7.1 Chapter overview

This chapter presents the main conclusion of this thesis. Section 7.2 re-emphasizes the research aim, the primary biomedical signals used and the case studies explored. In section 7.3 the main contributions to the body of knowledge are presented regarding the technical and clinical knowledge advancements. Sections 7.4 and 7.5 report the answers to the research questions and the main objectives presenting a summary of findings and conclusions. Finally, section 7.6 highlights the main limitations met in the research and the roadmap for future work.

7.2 Research aim

The main aim of this thesis was to develop reliable and accurate frameworks and tools to monitor the relationship between the CVS and the ANS in real-life settings via biomedical signal processing and machine learning techniques to predict adverse healthcare events and automatically detect the onset of unhealthy risky situations.

As HRV is one of the best known biomedical signal, reliable and non-invasive tools available to monitor the relationship between the CVS and the ANS in real-life settings, it was selected as the main biomedical signal in this thesis. In support of that, HRV has been already used in several studies as a predictor of adverse healthcare events.

Mental stress and fall prediction in later-life were chosen as case studies as they are important problems for modern society. Moreover, it is known that mental stress causes alterations in both the CVS and the ANS and many wearable technologies attempt to detect stress in real-time, enhancing the interest in ultra-short

HRV analysis. Also, mental stress is easy to replicate in laboratory and real-life settings. On the other side, fall prediction in later-life is one of the best examples of rare healthcare events and it still presents many challenges to be addressed (e.g., technology- and algorithm-wise).

7.3 Contribution to the body of knowledge

This research produced novel results and a significant knowledge advancement for both the investigated health and wellbeing problems and as well as the technical and methodological approaches.

Regarding the technical and methodological knowledge advancements, this research was the first proposing a systematic approach to select ultra-short HRV features that are reliable surrogates of 5min HRV features. This result was achieved combining both standard statistical and machine learning methods. Since clear guidelines on ultra-short HRV analysis were not available, and clear methods to assess ultra-short HRV features were also missing, the presented framework was developed in alignment with the medical literature on surrogate outcomes [143, 144].

Moreover, a protocol more robust than the existing methods for the prediction of rare events (i.e., unbalanced datasets) was developed using machine learning algorithms. The framework was developed by splitting the whole dataset into three folders: for feature selection, for training and validating the classification models and for evaluating the performance of the developed classification models. Furthermore, to minimise the overfitting problem, a cross-validation procedure was repeated multiple times as the number of events to predict (i.e., falls) was the minority class.

Concerning the investigated health and wellbeing problems, this research proved that it is possible to automatically detect real mental stress with 1min recordings achieving sensitivity, specificity and accuracy rates of above 80%. This may seem a small achievement, but there is a recognisable difference in detecting acute mental stress in 1min or 5min during risky jobs (e.g., whilst performing neurosurgery, driving a truck or flying an aeroplane).

Moreover, as shown in Fig. 7.1, shortening HRV features below the standard 5min achieved better results than the study presented by Melillo *et al.* [45], which applied less robust machine learning techniques to the same dataset and used standard 5min HRV analysis. This suggested that HRV features computed in excerpts shorter than 5min are closer to the stress dynamics.

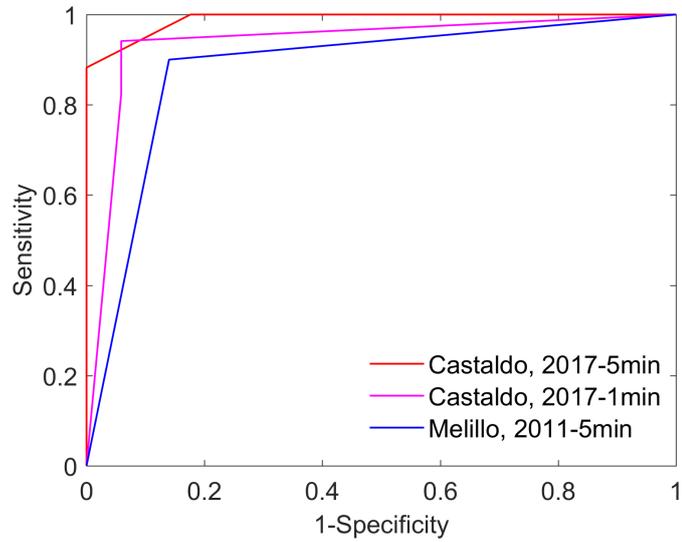


Figure 7.1: ROC curves comparison between Melillo *et al.* [45] and the study presented in this thesis to detect mental stress in real-life settings.

In relation to the prediction of falls in later-life, this research was the first demonstrated that short term HRV recordings could be used to identify future fallers with a sensitivity, specificity and accuracy rates of 72%, 61%, 68% respectively. This was the first time this causal dependency was demonstrated and several clinical studies are now testing those results on wider populations.

Moreover, as shown in Fig. 7.2, the study performed in this thesis achieved better results than the study presented by Melillo *et al.* [64], which applied to the same dataset less robust machine learning techniques (i.e., the dataset was not split into different folders for feature selection, training-validation and testing and the cross-validation procedure was not repeated multiple times to avoid bias) and used 24H HRV analysis (long HRV analysis).

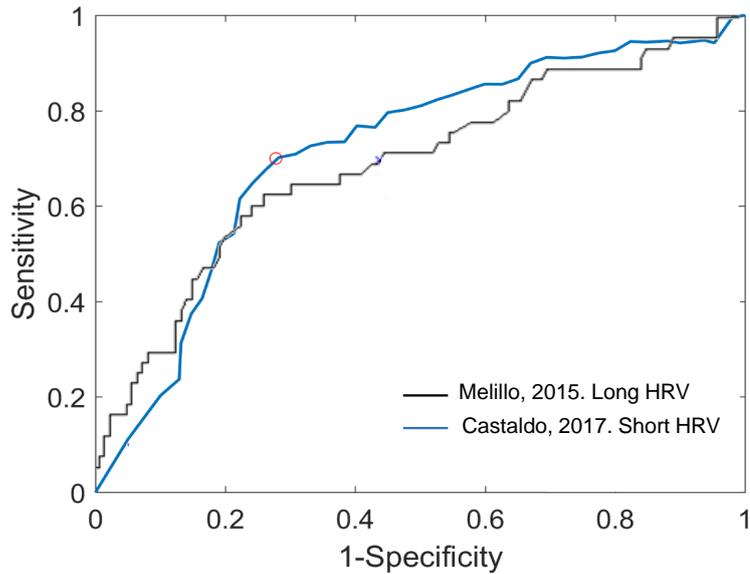


Figure 7.2: ROC curves comparison between Melillo *et al.* [64] and the study presented in this thesis to predict falls in later-life via HRV.

In conclusion, the results achieved in this thesis showed better results than previous studies due to the application of more structured frameworks to allow the translation of traditional laboratory methods of signal processing and machine learning techniques into real-life settings for the detection or prediction of adverse healthcare events.

7.4 Research questions and answers

Shifting healthcare monitoring techniques from the laboratory to real-life scenarios is very challenging. The current shift towards the use of advanced sensors in everyday objects (e.g., smartwatches) is strongly increasing the need for reliable methods and tools to analyse healthcare information acquired in real-life settings for wellbeing applications. Consequently, two research questions were explored and investigated:

Research Question 1: to what extent can the length of biomedical signals be shortened without losing their physiological meaning?

Research Question 2: how can current machine learning techniques be improved to reliably assess the interaction of the CVS and the ANS in real-life settings?

Regarding the first question, shortening physiological signals below the standard recommendations is of great interest in the scientific community as it may cause

a loss of reliability and accuracy in the detection or prediction of adverse healthcare events. Prior to the work performed in this thesis, few studies investigated the use of ultra-short term HRV and none of them proposed a reliable and accurate method to understand to what extent HRV features can be shortened below the standard recommendations. Previous standard methods found in the medical literature were based on correlation indices only, which suited descriptive statistics, but not predictive ones. Therefore, a new robust approach to explore to what extent ultra-short HRV features can be considered reliable surrogates of 5min HRV features was presented in this thesis. The aforementioned approach could also be used in other applications in which ultra-short and continuous monitoring of the CVS/ANS via wearable devices can be relevant (e.g., loss of attention in critical jobs.).

Regarding the second question, although many existing machine learning techniques to cope with small and unbalanced datasets are already in use in the existing literature, structured recommendations and guidelines to develop accurate machine learning algorithms to detect or predict adverse healthcare events using biomedical signals are still missing. Therefore, referring to existing machine learning techniques, recommendations and theoretical frameworks are suggested to reliably improve the assessment of the interaction of the CVS and the ANS using wearable sensors in small and unbalanced datasets.

7.5 Research objectives: summary of findings and conclusions

A summary of the main conclusions is presented in Table 7.1.

Table 7.1: Summary of the main results.

Case studies	Objectives	Main Conclusions	Thesis Chapters
CS1: mental stress detection	Obj1: method to assess ultra-short HRV	A robust framework to assess ultra-short HRV was developed based on the existing literature.	Chapter 3 & 4 & 5
	Obj2: ML for small dataset	A ML classifier was developed to automatically detect stress using 1 min HRV features achieving better performances than other methods in the literature.	Chapter 4 & 5
CS2: prediction of falls in later-life	Obj3: ML for unbalanced dataset	For the first time it was proved that short term HRV recordings can be used to identify future fallers. A structured framework for ML techniques to predict falls was developed.	Chapter 3 & 4 & 6

The first objective was to:

Objective 1: develop a novel approach to assess the reliability of biomedical signals length shorter than the standard recommendations in real-life settings.

In Chapter 4, a theoretical framework to select ultra-short HRV features that are reliable surrogates of 5min HRV features was proposed for the first time. This was accomplished by exploring the cardiovascular and autonomic response to mental stress in healthy subjects (Case Study 1). In Chapter 3, through a systematic review of the literature (deliverables 1a and 1b), it was demonstrated that little attention has been paid thus far to ultra-short term HRV analysis (i.e., less than 5 min) and no reliable methods were used to assess to what extent ultra-short HRV features could be used as surrogates of short ones. Consequently, in order to fill this gap and seek answers to the research questions, data were collected and analysed from three experiments: stress assessment in real life during an academic examination (AE); stress assessment in individual cognitive tasks (i.e., a Stroop Colour Test

(SCWT)); stress assessment in a group war scenario simulator (i.e., a war rescue mission in immersive and collaborative virtual gaming, VGC). For the first experiment, 42 healthy subjects were enrolled and monitored under two conditions: during an oral examination (i.e., a stress condition) and at resting after the aster holiday break. A robust protocol was applied consisting of: 1) extracting HRV features of different lengths (i.e., 5min (benchmark), 3min, 2min, 1min, 30sec) and applying statistical tests to observe whether the extracted HRV features were significantly different between groups (i.e., HRV features of the same length during rest and stress phases) and within groups (i.e., ultra-short HRV features versus 5min HRV features during rest and stress phases); 2) training and validating machine learning methods using 5min HRV features and testing them on ultra-short HRV features to verify if ultra-short HRV features can effectively be used to automatically detect mental stress (deliverables 1c and 1f). This study led to the development of a new robust framework to explore to what extent ultra-short HRV features can be considered reliable surrogates of 5min HRV features. This is currently of great interest as the continued rise of everyday wearable devices being able to instantaneously assess mental stress level is raising the attention of the scientific community around the use of HRV features computed over excerpts shorter than 5 minutes. This study proved that not all of the ultra-short HRV features were good surrogates of short term ones. In fact, only six ultra-short HRV features resulted in being good surrogates of short term ones: MeanNN, StdNN, MeanHR, StdHR, HF, and SD2. These six features displayed consistency across all the excerpt lengths (i.e., from 5 min to 1 min) and good performance if employed in a well dimensioned automatic classifier. In fact, an automatic classifier based on the K-nearest neighbours algorithm (IBK) was able to detect stressed subjects with very high performance, using 3min HRV analysis, and relatively good performance using 1min HRV excerpts. The former achieved sensitivity, specificity and accuracy of 94%, 94% and 94% respectively and the latter achieved 82% sensitivity, 94% specificity and 88% accuracy . Therefore, this suggested that it is possible to automatically detect mental stress using ultra-short HRV features with excerpts not shorter than 1 min. According to the specific application, 3 or 2 min excerpts could be preferable, because features having a clear physiological significance (e.g., HF and LF) remain computable. The model developed in this work could be applied to the situation of a mental effort such as an exam or job interview, that represents a long period under stress. Finally, it is useful to mention that the proposed methodology could be used in any application aiming to automatically detect a condition using ultra-short HRV features. In particular, the proposed method can improve the identification of the minimal length of HRV

excerpts required to enable the detection of an anomaly in quasi-real time.

However, due to the low number of subjects enrolled in this experiment, more experimental studies were designed and carried out to enrol more subjects with the aim to verify if ultra-short HRV feature may be useful in detecting stress. The choice of having laboratory experiments was due to the degree of control they provide in order to assess the relationship between acute mental stress and ultra-short term HRV analysis. However, the major disadvantage of experimental studies is that the nature of the experiment may differ from what people might actually do in everyday life. In fact, although the experiments were modelled as much as possible to simulate a real acute mental stress according to the existing literature, the effect of in-lab stressors resulted in them being less stressful than real-life stress. ECGs from 170 healthy subjects were acquired and analysed under rest and stress conditions (i.e., the SCWT and the VGC). This provided sufficient information to quantify the loss of performance using in-lab stressors and automatically detect mental stress using ultra-short signals acquired via wearable devices (deliverables 1d, 1c and 1f). The studies carried out in the laboratory environments demonstrated that it is possible to detect mental stress using ultra-short HRV features consolidating the results reported in the previous study. In fact, the six ultra-short HRV features (MeanNN, StdNN, MeanHR, StdHR, HF, and SD2) selected using a real stressor also showed to be significant during in-lab stress and they maintained the same trends for both real and in-lab stress. They also showed good discriminatory power when employed in a well-dimensioned classifier developed using in-lab stressors.

The second and third objectives were:

Objective 2: develop a pragmatic framework to improve machine learning techniques for small datasets.

Objective 3: develop a pragmatic framework to improve machine learning techniques for unbalanced datasets (i.e., reducing the number of false positive classifications and overfitting problems to predict rare events).

In Chapter 4, theoretical frameworks for small and unbalanced datasets were presented and in Chapters 5 and 6 the proposed frameworks were applied to the two specific case studies.

The proposed framework to improve machine learning techniques for small datasets was applied to Case Study 1, in particular to real-life stress. The dataset consisted of 42 healthy students undertaking a verbal academic examination. Although many algorithms already exist to cope with small datasets, the proposed

framework gathered together simple adjustments to improve the performance of classifiers and avoid overfitting problems. The problem of having a small dataset is often due to the scarcity of real data or a miscalculation of the minimum sample number in the study designs. In fact, in Chapter 5, section 5.3.1.1, a novel approach to calculate the minimum sample number for predictive models was proposed based on the study design and the desirable complexity of the model.

The approach proposed to improve machine learning algorithms for the prediction of rare events (i.e., unbalanced datasets) was applied to Case Study 2: “the cardiovascular and autonomic response to falls in later life”, which is a typical case of a rare, but severe, event that if not predicted in time may cause more severe clinical conditions. In Chapter 3, through a review of the main fall risks and existing technologies and tools to predict falls (deliverable 2a) it was evident that the use of HRV as a tool to predict falls in the elderly was still primordial in the scientific community and that there was the need for objective and clinically applicable methods to prevent and predict falls. Therefore, a study was carried out to address the existing gaps. The study consisted of a dataset of more than 4000 hours of continuous ECG recordings, acquired from 170 hypertensive patients (mean age above 55), of which 34 experienced an accidental fall (defined as an unintentionally coming to the ground or some lower level, not due to syncope) within three months from the recording after the baseline assessment was carried out. The most informative HRV features to predict falls in later-life were identified (deliverable 2b) and a model through rigorous training, validation and testing procedure, using three independent subsets of data for feature selection, model training and testing and averaging the performances by repeating the procedure ten times was developed to automatically predict falls (deliverable 2c). From the clinical point of view, this study proved that dysfunctions between the CVS and the ANS are associated with higher risk of falling. This study provided the first evidence that short term HRV recordings could be used to identify future fallers with a sensitivity, specificity and accuracy rates of 72%, 61%, 68% respectively. This was the first time this causal dependency was demonstrated and several clinical studies are now testing those results in wider populations [299, 300].

Overall, the development of basic methods and tools for enhancing the monitoring of CVS and ANS dynamics in real-life settings were applied to burgeoning problems: mental stress and accidental falls in later life, producing novel results and a significant knowledge advancement for both the investigated health and wellbeing problems and the technical and methodological ones.

7.6 Limitations and future work

This thesis provides a first attempt at proposing novel approaches to translate signal processing and data mining techniques from controlled environments (i.e., hospitals and research laboratories) into real-life settings using wearable sensors to detect or predict adverse healthcare events. In particular, novel frameworks to assess the validity of ultra-short HRV features and improve machine learning techniques for small and unbalanced datasets were explored. However, due to time and resource constraints, this research presents the following limitations.

In fact, although the developed approaches presented in this thesis could also be used in many healthcare applications, they were only applied to two specific problems (mental stress and accidental falls). Moreover, the validity of the proposed frameworks for small and unbalanced datasets was not compared to the existing methods. Nevertheless, for the two already acquired datasets: mental stress in real-life and falls in later-life, standard methods for small and unbalanced datasets were used by Melillo *et al.* [45, 64] and showed lower performances than the ones achieved in this research (Fig. 7.1 and 7.2).

This thesis also provides a first attempt to detect stress in healthy subjects using ultra-short HRV and predict falls in later-life using short HRV features.

Concerning the first case study (Fig. 7.3), ultra-short HRV features were not investigated in correlation with breathing rate, due to controversial concerns about the necessity to control breathing rate in relation to HRV features, however, since ultra-short HRV features have not been explored comprehensively and thoroughly, they should be assessed in relation to breathing rate and their validity investigated, especially in the frequency domain. Moreover, with the joint analysis of respiration and HRV, a more reliable characterisation of ANS response to stress could be obtained. Another factor to take into account toward the generalisation of the presented stress algorithm, is the investigation of ultra-short HRV features in different populations (e.g., older, with cardiovascular problems) and in different stressful situations (e.g., using a different kinds of stressors, such as physical, emotional, working stressors in real-life situations). Other aspects that should be evaluated to better detect stress are the intrinsic and extrinsic factors that could affect stress levels such as diet, quality of sleep and fitness levels. In regard to these, a first trial to investigate the quality of sleep associated with stress levels was carried out, however only 10 subjects were included in the trial and among them, only two subjects reported a poor quality of sleep, therefore, no conclusions could be drawn. Another important step for

the development of a general algorithm that could then be embedded in wearable devices is to associate HRV with other biomedical signals to achieve better accuracy than HRV alone. Indeed, actigraphy signals could also increase context awareness. In fact, analysis of physiological signals is more meaningful when presented along with situational context awareness which is necessary if the algorithms have to be embedded in wearable sensors. Moreover, they could discriminate between different stressors and help monitoring behavioural activities. Therefore, the final goal would be the development of an integrated wearable sensor to detect and monitor stress levels in risky situations or jobs.

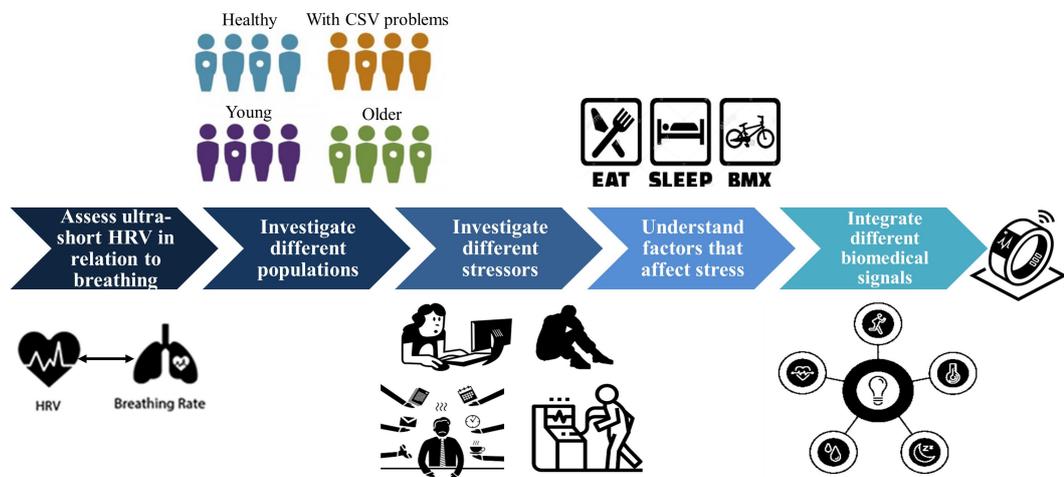


Figure 7.3: Roadmap for future work to improve mental stress detection. Several steps need to be assessed before generalising the proposed model to detect stress.

In relation to the second case study (Fig. 7.4), this presents some limitations that should be considered before adopting these methods in other contexts. This study was focused on hypertensive patients, which represent a special population with distinguished characteristics, different from the population of community-dwelling older citizens. Therefore, the association between short HRV features and the risk of falling should also be extended to a more general population, in particular, short HRV features should be investigated in patients with Parkinson's, cardiovascular problems and hospitalised patients. Moreover, exposure to other independent intrinsic/extrinsic risk factors for falls could be used to independently verify the results. Therefore, intrinsic factors such as visual impairment, gait and balance problems and sleep quality should be investigated using non-invasive biomedical signals (e.g., EMG, HRV). On the other hand, extrinsic factors such as a lack of stairs handrails, obstacles and tripping hazards and psychoactive medication

should also be investigated using cameras and actigraphy. Therefore, the final goal would be to develop an integrated wearable sensor to predict falls using vital signs to also control other risky situations that ageing could cause (e.g., strokes).

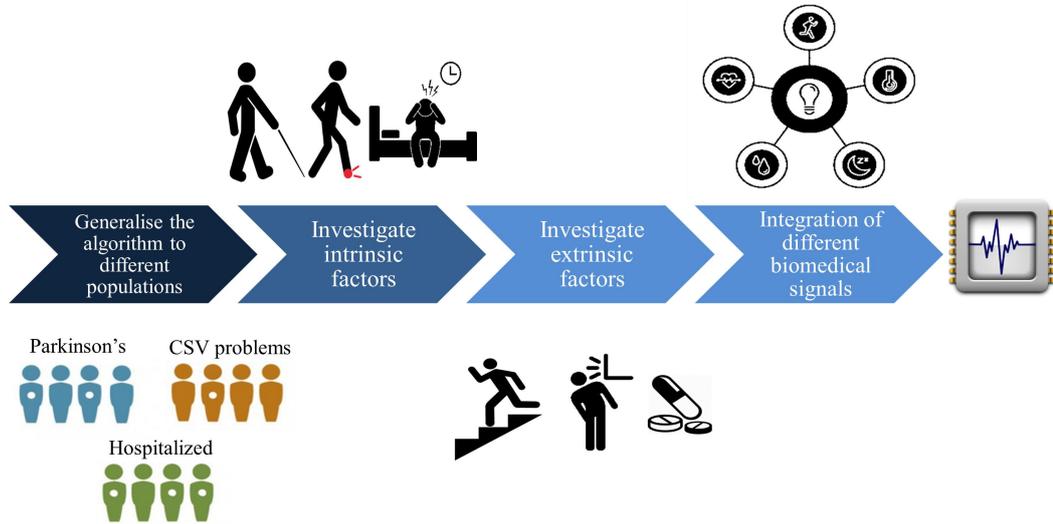


Figure 7.4: Roadmap for future work to improve fall prediction in later-life. Several steps need to be assessed before generalising the proposed model to predict falls.

References

- [1] Rossana Castaldo, Paolo Melillo, Umberto Bracale, M Caserta, Maria Triassi and Leandro Pecchia. “Acute mental stress assessment via short term HRV analysis in healthy adults: A systematic review with meta-analysis”. In: *Biomedical Signal Processing and Control* 18 (2015), pp. 370–377.
- [2] Rossana Castaldo, Paolo Melillo and Leandro Pecchia. “Acute mental stress assessment via short term HRV analysis in healthy adults: a systematic review”. In: *6th European Conference of the International Federation for Medical and Biological Engineering*. Vol. 45. Springer, Cham, Switzerland. 2015, pp. 1–4.
- [3] Leandro Pecchia, Rossana Castaldo, Luis Montesinos and Paolo Melillo. “Are ultra-short Heart Rate Variability features good surrogates of short term ones? Literature review and method recommendations”. In: *Healthcare Technology Letters* (2017). under review.
- [4] Rossana Castaldo, Paolo Melillo, Raffaele Izzo, Nicola De Luca and Leandro Pecchia. “Fall prediction in hypertensive patients via short-term HRV Analysis”. In: *IEEE Journal of Biomedical and Health Informatics* 21.2 (2017), pp. 399–406.
- [5] Rossana Castaldo, Luis Montesinos, Paolo Melillo and Leandro Pecchia. “Ultra-short term HRV Features as Surrogate of Short term HRV. A Case Study on Mental Stress Detection in Real Life”. In: *BMC Medical Informatics and Decision Making* (2017). Manuscript submitted for publication.
- [6] Rossana Castaldo, William Xu, Paolo Melillo, Leandro Pecchia, Lorena Santamaria and C James. “Detection of mental stress due to oral academic examination via ultra-short-term HRV analysis”. In: *Engineering in Medicine and Biology Society (EMBC), 2016. IEEE 38th Annual International Conference*. IEEE, Orlando, FL. 2016, pp. 3805–3808.
- [7] Rossana Castaldo, Paolo Melillo and Leandro Pecchia. “Acute Mental Stress Detection via Ultra-short term HRV Analysis”. In: *World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada*. Vol. 51. Springer, Cham, Switzerland. 2015, pp. 1068–1071.
- [8] Rossana Castaldo, Luis Montesinos, Paolo Melillo, Sebastiano Massaro and Leandro Pecchia. “To What Extent Can We Shorten HRV Analysis in Wearable Sensing? A Case Study on Mental Stress Detection.” In: *EMBECE & NBC 2017*. Vol. 65. Springer, 2017, pp. 643–646.
- [9] Rossana Castaldo, Luis Montesinos, Tim S Wan, Andra Serban, Sebastiano Massaro and Leandro Pecchia. “Heart Rate Variability Analysis and Performance during a Repeated Mental Workload Task”. In: *EMBECE & NBC 2017*. Vol. 65. Springer, 2017, pp. 69–72.

- [10] Rossana Castaldo and Leandro Pecchia. “Preliminary Results from a Proof of Concept Study for Fall Detection via ECG Morphology”. In: *XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016: MEDICON 2016, March 31st-April 2nd 2016, Paphos, Cyprus*. Vol. 57. Springer, Cham, Switzerland. 2016, p. 205.
- [11] Paolo Melillo, Rossana Castaldo, Giovanna Sannino, Ada Orrico, Giuseppe De Pietro and Leandro Pecchia. “Wearable technology and ECG processing for fall risk assessment, prevention and detection”. In: *Engineering in Medicine and Biology Society (EMBC), 2015. 37th Annual International Conference*. Vol. 2015. IEEE, Milan, Italy. 2015, pp. 7740–7743.
- [12] Shyamal Patel, Hyung Park, Paolo Bonato, Leighton Chan and Mary Rodgers. “A review of wearable sensors and systems with application in rehabilitation”. In: *Journal of Neuroengineering and Rehabilitation* 9.1 (2012), p. 21.
- [13] Arthur R Jensen and William D Rohwer. “The Stroop color-word test: a review”. In: *Acta psychologica* 25 (1966), pp. 36–93.
- [14] Ahmad Rauf Subahni, Likun Xia and Aamir Saeed Malik. “Association of mental stress with video games”. In: *Intelligent and Advanced Systems (ICIAS), 2012 4th International Conference on*. Vol. 1. IEEE. 2012, pp. 82–85.
- [15] Task Force. “Heart rate variability: standards of measurement, physiological interpretation and clinical use. Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology”. In: *Circulation* 93.5 (1996), pp. 1043–1065.
- [16] George Boateng and David Kotz. “StressAware: An App for Real-Time Stress Monitoring on the Amulet Wearable Platform”. In: *IEEE MIT Undergraduate Research Technology Conference (URTC)*. IEEE, Cambridge, MA, 2017.
- [17] *Tinke, Fitness Tracker for Exercise, Stress & Heart Rate — Zensorium*. Aug. 2016. URL: <https://www.zensorium.com/tinke/>.
- [18] *PulseOn*. Aug. 2016. URL: <http://pulseon.com/ohr>.
- [19] *NeuroSky*. Aug. 2016. URL: <http://neurosky.com/biosensors/ecg-sensor/algorithms/>.
- [20] *ithlete heart rate variability training tool*. Aug. 2016. URL: <https://www.myithlete.com/>.
- [21] Domenico Ribatti. “William Harvey and the discovery of the circulation of the blood”. In: *Journal of Angiogenesis Research* 1.1 (2009), p. 3.
- [22] Mark J Shen and Douglas P Zipes. “Role of the autonomic nervous system in modulating cardiac arrhythmias”. In: *Circulation Research* 114.6 (2014), pp. 1004–1021.
- [23] Harry A Fozzard, Edgar Haber, Robert B Jennings and Arnold M Katz. *The heart and Cardiovascular System*. Raven Press, 1986.
- [24] Purves D, Augustine GJ, Fitzpatrick D and et al. *Autonomic Regulation of Cardiovascular Function*. 2nd edition. Sunderland (MA): Sinauer Associates; Neuroscience., 2001.
- [25] Otto Appenzeller and Emilio Oribe. *The autonomic nervous system: an introduction to basic and clinical concepts*. Elsevier Health Sciences, Canada, USA, 1997.
- [26] Clifford B Saper. “The central autonomic nervous system: conscious visceral perception and autonomic pattern generation”. In: *Annual Review of Neuroscience* 25.1 (2002), pp. 433–469.

- [27] Agnieszka Zygmunt and Jerzy Stanczyk. “Methods of evaluation of autonomic nervous system function”. In: *Archives of Medical Science : AMS* 6.1 (2010), pp. 11–18.
- [28] Conny MA van Ravenswaaij-Arts, Louis AA Kollee, Jeroen CW Hopman, Gerard BA Stoeltinga and Herman P van Geijn. “Heart rate variability”. In: *Annals of Internal Medicine* 118.6 (1993), pp. 436–447.
- [29] Marek Malik. “Heart rate variability.” In: *Current Opinion in Cardiology* 13.1 (1998), pp. 36–44.
- [30] Gernot Ernst. *Heart Rate Variability*. SpringerVerlag, London, UK, 2016.
- [31] P Kamen. “Heart rate variability.” In: *Australian Family Physician* 25.7 (1996), pp. 1087–9.
- [32] Mika P Tarvainen, Juha-Pekka Niskanen, Jukka A Lipponen, Perttu O Ranta-Aho and Pasi A Karjalainen. “Kubios HRV–heart rate variability analysis software”. In: *Computer Methods and Programs in Biomedicine* 113.1 (2014), pp. 210–220.
- [33] Mika P Tarvainen, Juha-Pekka Niskanen, Jukka A Lipponen, Perttu O Ranta-Aho and Pasi A Karjalainen. “Kubios HRVheart rate variability analysis software”. In: *Computer Methods and Programs in Biomedicine* 113.1 (2014), pp. 210–220.
- [34] Clément Gallet and Claude Julien. “The significance threshold for coherence when using the Welch’s periodogram method: effect of overlapping segments”. In: *Biomedical Signal Processing and Control* 6.4 (2011), pp. 405–409.
- [35] Eduardo Miranda Dantas, Marcela Lima Sant’Anna, Rodrigo Varejão Andreão, Christine Pereira Gonçalves, Elis Aguiar Morra, Marcelo Perim Baldo, Sérgio Lamêgo Rodrigues and Jose Geraldo Mill. “Spectral analysis of heart rate variability with the autoregressive method: What model order to choose?” In: *Computers in Biology and Medicine* 42.2 (2012), pp. 164–170.
- [36] Ronald D Berger, Solange Akselrod, David Gordon and Richard J Cohen. “An efficient algorithm for spectral analysis of heart rate variability”. In: *IEEE Transactions on Biomedical Engineering* 9 (1986), pp. 900–904.
- [37] DS Fonseca, AD’Affonsêca Netto, RB Ferreira and AMFL Miranda de Sá. “Lomb-scargle periodogram applied to heart rate variability study”. In: *Biosignals and Biorobotics Conference (BRC), 2013 ISSNIP*. IEEE. 2013, pp. 1–4.
- [38] Fredric J Harris. “On the use of windows for harmonic analysis with the discrete Fourier transform”. In: *Proceedings of the IEEE* 66.1 (1978), pp. 51–83.
- [39] Elif Derya Übeyli, Hakan Işık and İnan Güler. “Application of FFT and arma spectral analysis to arterial doppler signals”. In: *Mathematical and Computational Applications* 8.3 (2003), pp. 311–318.
- [40] Sylvain Laborde, Emma Mosley and Julian F Thayer. “Heart rate variability and cardiac vagal tone in psychophysiological research–recommendations for experiment planning, data analysis, and data reporting”. In: *Frontiers in Psychology* 8 (2017), p. 213.
- [41] Aurélien Pichon, Manuel Roulaud, Sophie Antoine-Jonville, Claire de Bisschop and André Denjean. “Spectral analysis of heart rate variability: interchangeability between autoregressive analysis and fast Fourier transform”. In: *Journal of Electrocardiology* 39.1 (2006), pp. 31–37.

- [42] Gustavo A Reyes del Paso, Wolf Langewitz, Lambertus JM Mulder, Arie Roon and Stefan Duschek. “The utility of low frequency heart rate variability as an index of sympathetic cardiac tone: a review with emphasis on a reanalysis of previous studies”. In: *Psychophysiology* 50.5 (2013), pp. 477–487.
- [43] Zoltan German-Sallo. “Wavelet transform based HRV analysis”. In: *Procedia Technology* 12 (2014), pp. 105–111.
- [44] Luca T Mainardi. “On the quantification of heart rate variability spectral parameters using time–frequency and time-varying methods”. In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 367.1887 (2009), pp. 255–275.
- [45] Paolo Melillo, Marcello Bracale and Leandro Pecchia. “Nonlinear Heart Rate Variability features for real-life stress detection. Case study: students under stress due to university examination”. In: *Biomedical engineering online* 10.1 (2011), p. 96.
- [46] Michael Brennan, Marimuthu Palaniswami and Peter Kamen. “Do existing measures of Poincare plot geometry reflect nonlinear features of heart rate variability?” In: *IEEE transactions on Biomedical Engineering* 48.11 (2001), pp. 1342–1347.
- [47] Joshua S Richman and J Randall Moorman. “Physiological time-series analysis using approximate entropy and sample entropy”. In: *American Journal of Physiology-Heart and Circulatory Physiology* 278.6 (2000), H2039–H2049.
- [48] Raúl Carvajal, Niels Wessel, Montserrat Vallverdú, Pere Caminal and Andreas Voss. “Correlation dimension analysis of heart rate variability in patients with dilated cardiomyopathy”. In: *Computer Methods and Programs in Biomedicine* 78.2 (2005), pp. 133–140.
- [49] C-K Peng, Shlomo Havlin, H Eugene Stanley and Ary L Goldberger. “Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 5.1 (1995), pp. 82–87.
- [50] Joseph P Zbilut, Nitza Thomasson and Charles L Webber. “Recurrence quantification analysis as a tool for nonlinear exploration of nonstationary cardiac signals”. In: *Medical engineering & Physics* 24.1 (2002), pp. 53–60.
- [51] Steven M Pincus. “Approximate entropy as a measure of system complexity.” In: *Proceedings of the National Academy of Sciences* 88.6 (1991), pp. 2297–2301.
- [52] Kalon KL Ho, George B Moody, Chung-Kang Peng, Joseph E Mietus, Martin G Larson, Daniel Levy and Ary L Goldberger. “Predicting survival in heart failure case and control subjects by use of fully automated methods for deriving nonlinear and conventional indices of heart rate dynamics”. In: *Circulation* 96.3 (1997), pp. 842–848.
- [53] Manuela Ferrario, MARIA GABRIELLA Signorini and G Magenes. “Comparison between fetal heart rate standard parameters and complexity indexes for the identification of severe intrauterine growth restriction”. In: *Methods of Information in Medicine* 46.02 (2007), pp. 186–190.
- [54] Madalena Costa, Ary L Goldberger and C-K Peng. “Multiscale entropy analysis of complex physiologic time series”. In: *Physical Review Letters* 89.6 (2002), p. 068102.

- [55] C-K Peng, S Havlin, JM Hausdorff, JE Mietus, HE Stanley and AL Goldberger. “Fractal mechanisms and heart rate dynamics: long-range correlations and their breakdown with disease”. In: *Journal of Electrocardiology* 28 (1995), pp. 59–65.
- [56] Gari D Clifford. “Signal processing methods for heart rate variability”. PhD thesis. Department of Engineering Science, University of Oxford, 2002.
- [57] R Sinnreich, JD Kark, Y Friedlander, D Sapoznikov and MH Luria. “Five minute recordings of heart rate variability for population studies: repeatability and age–sex characteristics”. In: *Heart* 80.2 (1998), pp. 156–162.
- [58] Krishnan Muralikrishnan, Kabali Balasubramanian and Badanidiyur Viswanatha Rao. “Heart rate variability in normotensive subjects with family history of hypertension.” In: *Indian Journal of Applied Basic Medical Sciences* 55.3 (2011), pp. 253–261.
- [59] C Falcone, A Colonna, S Bozzini, B Matrone, L Guasti, EM Paganini, R Falcone and G Pelissero. “Cardiovascular risk factors and Sympatho-Vagal balance: Importance of time-domain heart rate variability”. In: *Journal of Clinical and Experimental Cardiology* 5.2 (2014).
- [60] JM Ryan and LG Howes. “Relations between alcohol consumption, heart rate, and heart rate variability in men”. In: *Heart* 88.6 (2002), pp. 641–642.
- [61] Massimo Pagani and Daniela Lucini. “Autonomic dysregulation in essential hypertension: insight from heart rate and arterial pressure variability”. In: *Autonomic Neuroscience* 90.1 (2001), pp. 76–82.
- [62] Hanna Mussalo, Esko Vanninen, Risto Ikäheimo, Tomi Laitinen, Markku Laakso, Esko Länsimies and Juha Hartikainen. “Heart rate variability and its determinants in patients with severe or mild essential hypertension”. In: *Clinical Physiology and Functional Imaging* 21.5 (2001), pp. 594–604.
- [63] Paolo Melillo, Raffaele Izzo, Nicola De Luca and Leandro Pecchia. “Heart rate variability and target organ damage in hypertensive patients”. In: *BMC Cardiovascular Disorders* 12.1 (2012), p. 105.
- [64] Paolo Melillo, Alan Jovic, Nicola De Luca and Leandro Pecchia. “Automatic classifier based on heart rate variability to identify fallers among hypertensive subjects”. In: *Healthcare Technology Letters* 2.4 (2015), pp. 89–94.
- [65] Ram Lochan Yadav, Prakash Kumar Yadav, Laxmi Kumari Yadav, Kopila Agrawal, Santosh Kumar Sah and Md Nazrul Islam. “Association between obesity and heart rate variability indices: an intuition toward cardiac autonomic alteration—a risk of CVD”. In: *Diabetes, Metabolic Syndrome and Obesity: Targets and Therapy* 10 (2017), p. 57.
- [66] Peretz Lavie and Anthony Trans Berris. *The enchanted world of sleep*. New Haven, CT, US: Yale University Press, 1996.
- [67] Narendra Singh, Dmitry Mironov, Paul W Armstrong, Allan M Ross, Anatoly Langer et al. “Heart rate variability assessment early after acute myocardial infarction”. In: *Circulation* 93.7 (1996), pp. 1388–1395.
- [68] Marcelo Risk, Vera Brill, Christopher Broadbridge and Alan Cohen. “Heart rate variability measurement in diabetic neuropathy: review of methods”. In: *Diabetes Technology & Therapeutics* 3.1 (2001), pp. 63–76.

- [69] Hadi Banaee, Mobyen Uddin Ahmed and Amy Loutfi. “Data mining for wearable sensors in health monitoring systems: a review of recent trends and challenges”. In: *Sensors* 13.12 (2013), pp. 17472–17500.
- [70] Hsun-Hsien Chang and José MF Moura. “Biomedical signal processing”. In: *Biomedical Engineering and Design Handbook 2* (2010), pp. 559–579.
- [71] Patrick Celka, Rolf Vetter, Philippe Renevey, Christophe Verjus, Victor Neuman, Jean Luprano, Jean-Dominique Decotignie and Christian Piguët. “Wearable biosensing: signal processing and communication architectures issues”. In: *Journal of Telecommunications and Information Technology* 4 (2005), pp. 90–104.
- [72] Ashwin Belle, Raghuram Thiagarajan, SM Sorousmehr, Fatemeh Navidi, Daniel A Beard and Kayvan Najarian. “Big data analytics in healthcare”. In: *BioMed Research International* 2015 (2015), pp. 370194–370210.
- [73] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), p. 436.
- [74] Ian H Witten, Eibe Frank, Mark A Hall and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, California, 2016.
- [75] Itamar Arel, Derek C Rose and Thomas P Karnowski. “Deep machine learning—a new frontier in artificial intelligence research [research frontier]”. In: *IEEE Computational Intelligence Magazine* 5.4 (2010), pp. 13–18.
- [76] Kenneth R Foster, Robert Koprowski and Joseph D Skufca. “Machine learning, medical diagnosis, and biomedical engineering research—commentary”. In: *Biomedical Engineering Online* 13.1 (2014), p. 94.
- [77] SJ Redmond, NH Lovell, GZ Yang, A Horsch, P Lukowicz, L Murrugarra and M Marschollek. “What does big data mean for wearable sensor systems?: Contribution of the IMIA wearable sensors in healthcare WG”. In: *Yearbook of Medical Informatics* 9.1 (2014), p. 135.
- [78] Vishakha Pandey and VK Giri. “High frequency noise removal from ECG using moving average filters”. In: *Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESSES), International Conference*. IEEE. 2016, pp. 191–195.
- [79] Gari D Clifford, Francisco Azuaje and Patrick Mcsharry. “ECG statistics, noise, artifacts, and missing data”. In: *Advanced Methods and Tools for ECG Data Analysis* 6 (2006), p. 18.
- [80] Ken Grauer. *A practical guide to ECG interpretation*. Mosby Incorporated, 1998.
- [81] Seema Nayak, MK Soni, Dipali Bansal et al. “Filtering techniques for ECG signal processing”. In: *Ijreas* 2.2 (2012), pp. 2249–3905.
- [82] P Morizet-Mahoudeaux, C Moreau, D Moreau and JJ Quarante. “Simple microprocessor-based system for on-line ECG arrhythmia analysis”. In: *Medical and Biological Engineering and Computing* 19.4 (1981), pp. 497–500.
- [83] J Fraden and MR Neuman. “QRS wave detection”. In: *Medical and Biological Engineering and computing* 18.2 (1980), pp. 125–132.
- [84] D Gustafson et al. “Automated VCG interpretation studies using signal analysis techniques”. In: *R-1044 Charles Stark Draper Lab., Cambridge, MA* (1977), p. 30.

- [85] A Menrad et al. “Dual microprocessor system for cardiovascular data acquisition, processing and recording”. In: *Proc. 1981 IEEE Int. Conf. Industrial Elect. Contr. Instrument.* 1981, pp. 64–69.
- [86] William P Holsinger, Kenneth M Kempner and Martin H Miller. “A QRS preprocessor based on digital differentiation”. In: *IEEE Transactions on Biomedical Engineering* 18.3 (1971), pp. 212–217.
- [87] RA Balda, G Diller, E Deardorff, J Doue and P Hsieh. “The HP ECG analysis program”. In: *Trends in Computer-Processed Electrocardiograms* 4 (1977), pp. 197–205.
- [88] Mark L Ahlstrom and Willis J Tompkins. “Automated high-speed analysis of Holter tapes with microcomputers”. In: *IEEE Transactions on Biomedical Engineering* 30.10 (1983), pp. 651–657.
- [89] WAH Engelse and C Zeelenberg. “A single scan algorithm for QRS-detection and feature extraction”. In: *Computers in Cardiology* 6.1979 (1979), pp. 37–42.
- [90] Masahiko Okada. “A digital filter for the QRS complex detection”. In: *IEEE Transactions on Biomedical Engineering* 26.12 (1979), pp. 700–703.
- [91] M Malik. “Effect of electrocardiogram recognition artifact on time-domain measurement of heart rate variability”. In: *Heart Rate Variability*. Armonk, NY: Futura Publishing Co., Inc (1995), pp. 99–118.
- [92] Mirja Peltola. “Role of editing of RR intervals in the analysis of heart rate variability”. In: *Frontiers in physiology* 3 (2012), p. 148.
- [93] Karl Pearson. “Principal components analysis”. In: *The London, Edinburgh and Dublin Philosophical Magazine and Journal* 6.2 (1901), p. 566.
- [94] Stephen Roberts and Richard Everson. *Independent component analysis: principles and practice*. Cambridge University Press, 2001.
- [95] Ron Kohavi and George H John. “Wrappers for feature subset selection”. In: *Artificial Intelligence* 97.1-2 (1997), pp. 273–324.
- [96] Marina Sokolova and Guy Lapalme. “A systematic analysis of performance measures for classification tasks”. In: *Information Processing & Management* 45.4 (2009), pp. 427–437.
- [97] Tom Fawcett. “An introduction to ROC analysis”. In: *Pattern recognition letters* 27.8 (2006), pp. 861–874.
- [98] John A Swets, Robyn M Dawes and John Monahan. “Better decisions through science”. In: *Scientific American* 283.4 (2000), pp. 82–87.
- [99] John A Swets. “Measuring the accuracy of diagnostic systems”. In: *Science* 240.4857 (1988), pp. 1285–1293.
- [100] Karimollah Hajian-Tilaki. “Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation”. In: *Caspian Journal of Internal Medicine* 4.2 (2013), p. 627.
- [101] David J Hand. “Measuring classifier performance: a coherent alternative to the area under the ROC curve”. In: *Machine learning* 77.1 (2009), pp. 103–123.
- [102] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine Learning* 20.3 (1995), pp. 273–297.

- [103] Fei Hu, Meng Jiang, Laura Celentano and Yang Xiao. “Robust medical ad hoc sensor networks (MASN) with wavelet-based ECG data mining”. In: *Ad Hoc Networks* 6.7 (2008), pp. 986–1012.
- [104] Mukta Paliwal and Usha A Kumar. “Neural Networks and Statistical Techniques: A review of Applications”. In: *Expert systems with applications* 36.1 (2009), pp. 2–17.
- [105] Thi Hong Nhan Vu, Namkyu Park, Yang Koo Lee, Yongmi Lee, Jong Yun Lee and Keun Ho Ryu. “Online discovery of Heart Rate Variability patterns in mobile healthcare services”. In: *Journal of Systems and Software* 83.10 (2010), pp. 1930–1940.
- [106] Christos A Frantzidis, Charalampos Bratsas, Manousos A Klados, Evdokimos Konstantinidis, Chrysa D Lithari, Ana B Vivas, Christos L Papadelis, Eleni Kaldoudi, Costas Pappas and Panagiotis D Bamidis. “On the classification of emotional biosignals evoked while viewing affective pictures: an integrated data-mining-based approach for healthcare applications”. In: *IEEE Transactions on Information Technology in Biomedicine* 14.2 (2010), pp. 309–318.
- [107] Irina Rish. “An empirical study of the naive Bayes classifier”. In: *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*. Vol. 3. 22. IBM. 2001, pp. 41–46.
- [108] Igor Kononenko. “Inductive and Bayesian learning in medical diagnosis”. In: *Applied Artificial Intelligence an International Journal* 7.4 (1993), pp. 317–337.
- [109] Mark Girolami and Simon Rogers. “Variational Bayesian multinomial probit regression with Gaussian process priors”. In: *Neural Computation* 18.8 (2006), pp. 1790–1817.
- [110] Nir Friedman, Dan Geiger and Moises Goldszmidt. “Bayesian network classifiers”. In: *Machine Learning* 29.2-3 (1997), pp. 131–163.
- [111] João Maroco, Dina Silva, Ana Rodrigues, Manuela Guerreiro, Isabel Santana and Alexandre de Mendonça. “Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests”. In: *BMC research notes* 4.1 (2011), p. 299.
- [112] S Vijayarani and M Muthulakshmi. “Comparative analysis of bayes and lazy classification algorithms”. In: *International Journal of Advanced Research in Computer and Communication Engineering* 2.8 (2013), pp. 3118–3124.
- [113] Stephen J Preece, John Y Goulermas, Laurence PJ Kenney, Dave Howard, Kenneth Meijer and Robin Crompton. “Activity identification using body-mounted sensors a review of classification techniques”. In: *Physiological measurement* 30.4 (2009), R1.
- [114] Richard J Gerrig and Philip G Zimbardo. *American Psychological Association: Glossary of Psychological Terms*. Pearson Education, Education, Incorporated (COR), 2002.
- [115] Susan Folkman. *Stress: Appraisal and Coping*. Springer, New York, USA, 2013.
- [116] Ora Kofman, Nachshon Meiran, Efrat Greenberg, Meirav Balas and Hagit Cohen. “Enhanced performance on executive functions associated with examination stress: Evidence from task-switching and Stroop paradigms”. In: *Cognition & Emotion* 20.5 (2006), pp. 577–595.

- [117] Tarani Chandola, Alexandros Heraclides and Meena Kumari. “Psychophysiological biomarkers of workplace stressors”. In: *Neuroscience & Biobehavioral Reviews* 35.1 (2010), pp. 51–57.
- [118] Rajiv Ranjan Singh, Sailesh Conjeti and Rahul Banerjee. “A comparative evaluation of neural network classifiers for stress level analysis of automotive drivers using physiological signals”. In: *Biomedical Signal Processing and Control* 8.6 (2013), pp. 740–754.
- [119] Suzanne C Segerstrom and Gregory E Miller. “Psychological stress and the human immune system: a meta-analytic study of 30 years of inquiry.” In: *Psychological Bulletin* 130.4 (2004), p. 601.
- [120] Nis Hjortskov, Dag Rissén, Anne Katrine Blangsted, Nils Fallentin, Ulf Lundberg and Karen Søgaard. “The effect of mental stress on heart rate variability and blood pressure during computer work”. In: *European journal of applied physiology* 92.1-2 (2004), pp. 84–89.
- [121] Vesna Vuksanović and Vera Gal. “Heart rate variability in mental stress aloud”. In: *Medical engineering & Physics* 29.3 (2007), pp. 344–349.
- [122] Zhibin Li, Harold Snieder, Shaoyong Su, Xiuhua Ding, Julian F Thayer, Frank A Treiber and Xiaoling Wang. “A longitudinal study in youth of heart rate variability at rest and in response to stress”. In: *International Journal of Psychophysiology* 73.3 (2009), pp. 212–217.
- [123] C Schubert, M Lambertz, RA Nelesen, W Bardwell, J-B Choi and JE Dimsdale. “Effects of stress on heart rate complexity: a comparison between short-term and chronic stress”. In: *Biological Psychology* 80.3 (2009), pp. 325–332.
- [124] Elizabeth Tharion, Sangeetha Parthasarathy and Nithya Neelakantan. “Short-term heart rate variability measures in students during examinations.” In: *Natl Med J India* 22.2 (2009), pp. 63–66.
- [125] Helmut Karl Lackner, Ilona Papousek, Jerry Joseph Batzel, Andreas Roessler, Hermann Scharfetter and Helmut Hinghofer-Szalkay. “Phase synchronization of hemodynamic variables and respiration during mental challenge”. In: *International Journal of Psychophysiology* 79.3 (2011), pp. 401–409.
- [126] Ilona Papousek, Karin Nauschnegg, Manuela Paechter, Helmut K Lackner, Nandu Goswami and Günter Schuster. “Trait and state positive affect and cardiovascular recovery from experimental academic stress”. In: *Biological Psychology* 83.2 (2010), pp. 108–115.
- [127] Joachim Taelman, Steven Vandeput, Elke Vlemincx, Arthur Spaepen and Sabine Van Huffel. “Instantaneous changes in heart rate regulation due to mental load in simulated office work”. In: *European Journal of Applied Physiology* 111.7 (2011), pp. 1497–1505.
- [128] Russo G C. A. Traina M. Galullo F. “Effects. of anxiety due to mental stress on heart rate variability in healthy subjects”. In: *Minerva Psichiatr* 52.4 (2011), pp. 227–31.
- [129] Z Visnovcova, M Mestanik, M Javorka, D Mokra, M Gala, A Jurko, A Calkovska and I Tonhajzerova. “Complexity and time asymmetry of heart rate variability are altered in acute mental stress”. In: *Physiological Measurement* 35.7 (2014), p. 1319.
- [130] J Martin Bland and Douglas G Altman. “Transformations, means, and confidence intervals.” In: *BMJ: British Medical Journal* 312.7038 (1996), p. 1079.

- [131] Alex J Sutton, Keith R Abrams, David R Jones, Trevor A Sheldon and Fujian Song. “Methods for meta-analysis in medical research”. In: *Statistics in Medicine* 22.19 (2003), pp. 3111–3114.
- [132] John A Chalmers, Daniel S Quintana, Maree J Abbott, Andrew H Kemp et al. “Anxiety disorders are associated with reduced heart rate variability: a meta-analysis”. In: *Frontiers in Psychiatry* 5 (2014), p. 80.
- [133] *OpenMeta[Analyst]*. Aug. 2015. URL: <http://www.cebm.brown.edu/openmeta/>.
- [134] Sidney Siegel. *Nonparametric Statistics for the Behavioral Sciences*. McGraw-hill, 1956.
- [135] H Stefan Bracha. “Freeze, flight, fight, fright, faint: adaptationist perspectives on the acute stress response spectrum.” In: *CNS Spectrums* 9.09 (2004), pp. 679–685.
- [136] Satya A Paritala. “Effects of physical and mental tasks on heart rate variability”. PhD thesis. Kakatiya University, India, 2009.
- [137] Ki H Chon, Christopher G Scully and Sheng Lu. “Approximate entropy for all signals”. In: *IEEE Engineering in Medicine and Biology Magazine* 28.6 (2009).
- [138] Karl Pearson. “Notes on the history of correlation”. In: *Biometrika* 13.1 (1920), pp. 25–45.
- [139] Xiaolin Yu and Jianbao Zhang. “Estimating the cortex and autonomic nervous activity during a mental arithmetic task”. In: *Biomedical Signal Processing and Control* 7.3 (2012), pp. 303–308.
- [140] Subramanya Mayya, Vivek Jilla, Vijay Narayan Tiwari, Mithun Manjnath Nayak and Rangavital Narayanan. “Continuous monitoring of stress on smartphone using heart rate variability”. In: *Bioinformatics and Bioengineering (BIBE), 2015. IEEE 15th International Conference*. IEEE, Washington, DC. 2015, pp. 1–5.
- [141] Ulrich Reimer, Emanuele Laurenzi, Edith Maier and Tom Ulmer. “Mobile Stress Recognition and Relaxation Support with SmartCoping: User-Adaptive Interpretation of Physiological Stress Parameters”. In: *Proceedings of the 50th Hawaii International Conference on System Sciences*. 2017.
- [142] Jennifer M Yentes, Nathaniel Hunt, Kendra K Schmid, Jeffrey P Kaipust, Denise McGrath and Nicholas Stergiou. “The appropriate use of approximate entropy and sample entropy with short data sets”. In: *Annals of Biomedical Engineering* 41.2 (2013), pp. 349–365.
- [143] Thomas R Fleming and David L DeMets. “Surrogate end points in clinical trials: are we being misled?” In: *Annals of Internal Medicine* 125.7 (1996), pp. 605–613.
- [144] Lucas Gallo, Cagla Eskicioglu, Luis H Braga, Forough Farrokhyar and Achilleas Thoma. “Users guide to the surgical literature: how to assess an article using surrogate end points”. In: *Canadian Journal of Surgery* 60.4 (2017), p. 280.
- [145] A Arza, JM Garzón, Alberto Hemando, Jordi Aguiló and Raquel Bailón. “Towards an objective measurement of emotional stress: Preliminary analysis based on heart rate variability”. In: *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. Vol. 2015. IEEE, Milano, Italy. 2015, pp. 3331–3334.
- [146] Hyun Jae Baek, Chul-Ho Cho, Jaegool Cho and Jong-Min Woo. “Reliability of ultra-short-term analysis as a surrogate of standard 5-min analysis of heart rate variability”. In: *Telemedicine and e-Health* 21.5 (2015), pp. 404–414.

- [147] Sansanee Boonnithi and Sukanya Phongsuphap. “Comparison of heart rate variability measures for mental stress detection”. In: *Computing in Cardiology, 2011*. IEEE. 2011, pp. 85–88.
- [148] Donatella Brisinda, Angela Venuti, Claudia Cataldi, Kristian Efremov, Emilia Intorno and Riccardo Fenici. “Real-time Imaging of Stress-induced Cardiac Autonomic Adaptation During Realistic Force-on-force Police Scenarios”. In: *Journal of Police and Criminal Psychology* 30.2 (2015), pp. 71–86.
- [149] Jongyoon Choi and Ricardo Gutierrez-Osuna. “Using heart rate monitors to detect mental stress”. In: *Wearable and Implantable Body Sensor Networks, 2009. Sixth International Workshop*. IEEE. 2009, pp. 219–223.
- [150] M De Rivecourt, MN Kuperus, WJ Post and LJM Mulder. “Cardiovascular and eye activity measures as indices for momentary changes in mental effort during simulated flight”. In: *Ergonomics* 51.9 (2008), pp. 1295–1319.
- [151] Michael R Esco and Andrew A Flatt. “Ultra-short-term heart rate variability indexes at rest and post-exercise in athletes: evaluating the agreement with accepted recommendations”. In: *Journal of Sports Science & Medicine* 13.3 (2014), p. 535.
- [152] Andrew A Flatt and Michael R Esco. “Validity of the athlete™ smart phone application for determining ultra-short-term heart rate variability”. In: *Journal of human kinetics* 39.1 (2013), pp. 85–92.
- [153] Desok Kim, Yunhwan Seo, Jaegeol Cho and Chul-Ho Cho. “Detection of subjects with higher self-reporting stress scores using heart rate variability patterns during the day”. In: *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. Vol. 2008. IEEE. 2008, pp. 682–685.
- [154] Sungjun Kwon, Dongseok Lee, Jeehoon Kim, Youngki Lee, Seungwoo Kang, Sangwon Seo and Kwangsuk Park. “Sinabro: A Smartphone-Integrated Opportunistic Electrocardiogram Monitoring System”. In: *Sensors* 16.3 (2016), p. 361.
- [155] James McNames and Mateo Aboy. “Reliability and accuracy of heart rate variability metrics versus ECG segment duration”. In: *Medical and Biological Engineering and Computing* 44.9 (2006), pp. 747–756.
- [156] M Loretto Munoz, Arie van Roon, Harriëtte Riese, Chris Thio, Emma Oostenbroek, Iris Westrik, Eco JC de Geus, Ron Gansevoort, Joop Lefrandt, Ilja M Nolte et al. “Validity of (ultra-) short recordings for heart rate variability measurements”. In: *PLoS One* 10.9 (2015), e0138921.
- [157] Mimma Nardelli, Alberto Greco, Juan Bolea, Gaetano Valenza, Enzo Pasquale Scilingo and Raquel Bailon. “Reliability of Lagged Poincaré Plot parameters in ultra-short Heart Rate Variability series: Application on Affective Sounds”. In: *IEEE Journal of Biomedical and Health Informatics* PP.99 (2017).
- [158] Udi Nussinovitch, Keren Politi Elishkevitz, Keren Katz, Moshe Nussinovitch, Shlomo Segev, Benjamin Volovitz and Naomi Nussinovitch. “Reliability of ultra-short ECG indices for heart rate variability”. In: *Annals of Noninvasive Electrocardiology* 16.2 (2011), pp. 117–122.
- [159] Parul Pandey, Eun Kyung Lee and Dario Pompili. “A Distributed Computing Framework for Real-time Detection of Stress and of its Propagation in a Team”. In: *IEEE Journal of Biomedical and Health Informatics* 20.6 (2016), pp. 1502–1512.

- [160] Tânia Pereira, Pedro R Almeida, João PS Cunha and Ana Aguiar. “Heart rate variability metrics for fine-grained stress level assessment”. In: *Computer Methods and Programs in Biomedicine* 148 (2017), pp. 71–80.
- [161] Lizawati Salahuddin, Jaegeol Cho, Myeong Gi Jeong and Desok Kim. “Ultra short term analysis of heart rate variability for monitoring mental stress in mobile settings”. In: *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*. Vol. 2007. IEEE, Lyon, France. 2007, pp. 4656–4659.
- [162] Lizawati Salahuddin, Myeong Gi Jeong and Desok Kim. “Ultra short term analysis of heart rate variability using normal sinus rhythm and atrial fibrillation ECG data”. In: *e-Health Networking, Application and Services, 2007 9th International Conference*. IEEE, Cork, Ireland. 2007, pp. 240–243.
- [163] Emily B Schroeder, Eric A Whitsel, Gregory W Evans, Ronald J Prineas, Lloyd E Chambless and Gerardo Heiss. “Repeatability of heart rate variability measures”. In: *Journal of Electrocardiology* 37.3 (2004), pp. 163–172.
- [164] Feng-Tso Sun, Cynthia Kuo, Heng-Tze Cheng, Senaka Butthpitiya, Patricia Collins and Martin Griss. “Activity-aware mental stress detection using physiological sensors”. In: *International Conference on Mobile Computing, Applications, and Services*. Vol. 76. Springer, 2010, pp. 282–301.
- [165] Tran Thong, Kehai Li, James McNames, Mateo Aboy and Brahm Goldstein. “Accuracy of ultra-short heart rate variability measures”. In: *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*. Vol. 3. IEEE, Cancn, Mexico. 2003, pp. 2424–2427.
- [166] Jacqueline Wijsman, Bernard Grundlehner, Hao Liu, Hermie Hermens and Julien Penders. “Towards mental stress detection using wearable physiological sensors”. In: *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, Boston, MA. 2011, pp. 1798–1801.
- [167] Qianli Xu, Tin Lay Nwe and Cuntai Guan. “Cluster-based analysis for personalized stress evaluation using physiological signals”. In: *IEEE Journal of Biomedical and Health Informatics* 19.1 (2015), pp. 275–281.
- [168] Xiaoling Wang, Xiuhua Ding, Shaoyong Su, Zhibin Li, Harriette Riese, Julian F Thayer, Frank Treiber and Harold Snieder. “Genetic influences on heart rate variability at rest and during stress”. In: *Psychophysiology* 46.3 (2009), pp. 458–465.
- [169] William R Rice. “Analyzing tables of statistical tests”. In: *Evolution* 43.1 (1989), pp. 223–225.
- [170] Parampreet Kaur, Jill C Stoltzfus et al. “Bland–Altman plot: A brief overview”. In: *International Journal of Academic Medicine* 3.1 (2017), p. 110.
- [171] J Cohen. “Statistical power analysis for the behavioral sciences, Stat”. In: *Power Anal. Behav. Sci.* 2nd 567 (1988).
- [172] Joseph P Weir. “Quantifying test-retest reliability using the intraclass correlation coefficient and the SEM”. In: *Journal of Strength and Conditioning Research* 19.1 (2005), p. 231.

- [173] Nienke M Kosse, Kim Brands, Jurgen M Bauer, Tibor Hortobágyi and Claudine JC Lamoth. “Sensor technologies aiming at fall prevention in institutionalized old adults: A synthesis of current knowledge”. In: *International journal of medical informatics* 82.9 (2013), pp. 743–752.
- [174] National Institute for Clinical Excellence, Great Britain et al. *Falls: the assessment and prevention of falls in older people*. National Institute for Clinical Excellence, 2004.
- [175] Isis Montalvo. “The national database of nursing quality indicators”. In: *OJIN: The Online Journal of Issues in Nursing* 12.3 (2007), pp. 112–214.
- [176] World Health Organization et al. “Ageing; Life Course Unit. WHO global report on falls prevention in older age”. In: *World Health Organization* (2008).
- [177] Paula C Fletcher and John P Hirdes. “Risk factors for falling among community-based seniors using home care services”. In: *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences* 57.8 (2002), pp. M504–M510.
- [178] Pekka Kannus, Harri Sievänen, Mika Palvanen, Teppo Järvinen and Jari Parkkari. “Prevention of falls and consequent injuries in elderly people”. In: *The Lancet* 366.9500 (2005), pp. 1885–1893.
- [179] Teresa M Steffen, Timothy A Hacker and Louise Mollinger. “Age-and gender-related test performance in community-dwelling elderly people: Six-Minute Walk Test, Berg Balance Scale, Timed Up & Go Test, and gait speeds”. In: *Physical Therapy* 82.2 (2002), pp. 128–137.
- [180] Susan W Muir, Katherine Berg, Bert Chesworth and Mark Speechley. “Use of the Berg Balance Scale for predicting multiple falls in community-dwelling elderly people: a prospective study”. In: *Physical Therapy* 88.4 (2008), pp. 449–459.
- [181] Michel Raïche, Réjean Hébert, François Prince and Hélène Corriveau. “Screening older adults at risk of falling with the Tinetti balance scale”. In: *The Lancet* 356.9234 (2000), pp. 1001–1002.
- [182] Janice M Morse et al. *Preventing Patient Falls*. Springer Publishing Company, New York, USA, 2008.
- [183] Diane M Wrisley, Gregory F Marchetti, Diane K Kuharsky and Susan L Whitney. “Reliability, internal consistency, and validity of data obtained with the functional gait assessment”. In: *Physical Therapy* 84.10 (2004), pp. 906–918.
- [184] Fay B Horak, Diane M Wrisley and James Frank. “The balance evaluation systems test (BESTest) to differentiate balance deficits”. In: *Physical Therapy* 89.5 (2009), pp. 484–498.
- [185] David Oliver, M Britton, P Seed, FC Martin and AH Hopper. “Development and evaluation of evidence based risk assessment tool (STRATIFY) to predict which elderly inpatients will fall: case-control and cohort studies”. In: *Bmj* 315.7115 (1997), pp. 1049–1053.
- [186] DG Gray-Miceli. “Fall risk assessment for older adults: the Hendrich II model”. In: *Annals of Long-Term Care* 15.2 (2007), pp. 1524–7929.

- [187] Heike A Bischoff, Hannes B Stähelin, Andreas U Monsch, Maura D Iversen, Antje Weyh, Margot Von Dechend, Regula Akos, Martin Conzelmann, Walter Dick and Robert Theiler. “Identifying a cut-off point for normal mobility: a comparison of the timed up and go test in community-dwelling and institutionalised elderly women”. In: *Age and ageing* 32.3 (2003), pp. 315–320.
- [188] Mary A Murphy, Sharon L Olson, Elizabeth J Protas and Averell R Overby. “Screening for falls in community-dwelling elderly”. In: *Journal of Aging and Physical Activity* 11.1 (2003), pp. 66–80.
- [189] Erika Jonsson, Marketta Henriksson and Helga Hirschfeld. “Does the functional reach test reflect stability limits in elderly people?” In: *Journal of Rehabilitation Medicine* 35.1 (2003), pp. 26–30.
- [190] Astrid Bergland and Knut Laake. “Concurrent and predictive validity of getting up from lying on the floor”. In: *Aging Clinical and Experimental Research* 17.3 (2005), pp. 181–185.
- [191] Bruno J Vellas, Sharon J Wayne, Linda Romero, Richard N Baumgartner, Laurence Z Rubenstein and Philip J Garry. “One-leg balance is an important predictor of injurious falls in older persons”. In: *Journal of the American Geriatrics Society* 45.6 (1997), pp. 735–738.
- [192] Lillemor Lundin-Olsson, Lars Nyberg, Yngve Gustafson et al. “Stops walking when talking as a predictor of falls in elderly people”. In: *Lancet* 349.9052 (1997), p. 617.
- [193] Marta Aranda-Gallardo, Jose M Morales-Asencio, Jose C Canca-Sanchez, Silvia Barrero-Sojo, Claudia Perez-Jimenez, Angeles Morales-Fernandez, Margarita Enriquez de Luna-Rodriguez, Ana B Moya-Suarez and Ana M Mora-Banderas. “Instruments for assessing the risk of falls in acute hospitalized patients: a systematic review and meta-analysis”. In: *BMC Health Services Research* 13.1 (2013), p. 122.
- [194] Ian D Cameron, Geoff R Murray, Lesley D Gillespie, M Clare Robertson, Keith D Hill, Robert G Cumming, Ngaire Kerse et al. “Interventions for preventing falls in older people in nursing care facilities and hospitals”. In: *Cochrane Database Syst Rev* 1.1 (2010).
- [195] Emma Barry, Rose Galvin, Claire Keogh, Frances Horgan and Tom Fahey. “Is the Timed Up and Go test a useful predictor of risk of falls in community dwelling older adults: a systematic review and meta-analysis”. In: *BMC Geriatrics* 14.1 (2014), p. 14.
- [196] Anne Tiedemann, Hiroyuki Shimada, Catherine Sherrington, Susan Murray and Stephen Lord. “The comparative ability of eight functional mobility tests for predicting falls in community-dwelling older people”. In: *Age and ageing* 37.4 (2008), pp. 430–435.
- [197] Severine Buatois, Darko Miljkovic, Patrick Manckoundia, Rene Gueguen, Patrick Miget, Guy Vançon, Philippe Perrin and Athanase Benetos. “Five times sit to stand test is a predictor of recurrent falls in healthy community-living subjects aged 65 and older”. In: *Journal of the American Geriatrics Society* 56.8 (2008), pp. 1575–1577.
- [198] Katherine Berg, Sharon Wood-Dauphine, JI Williams and David Gayton. “Measuring balance in the elderly: preliminary development of an instrument”. In: *Physiotherapy Canada* 41.6 (1989), pp. 304–311.
- [199] FJ Imms and OG Edholm. “Studies of gait and mobility in the elderly”. In: *Age and ageing* 10.3 (1981), pp. 147–156.

- [200] Jack M Guralnik, Eleanor M Simonsick, Luigi Ferrucci, Robert J Glynn, Lisa F Berkman, Dan G Blazer, Paul A Scherr and Robert B Wallace. “A short physical performance battery assessing lower extremity function: association with self-reported disability and prediction of mortality and nursing home admission”. In: *Journal of gerontology* 49.2 (1994), pp. M85–M94.
- [201] Mary Jo Gibson. “The prevention of falls in later life-a report of the Kellogg International Work Group on the prevention of falls by the elderly”. In: *Danish Medical Bulletin* 34.14 (1987), pp. 1–24.
- [202] Gerwin AL Meijer, Klaas R Westerterp, Francois MH Verhoeven, Hans BM Koper and Foppe ten Hoor. “Methods to assess physical activity with special reference to motion sensors and accelerometers”. In: *IEEE Transactions on Biomedical Engineering* 38.3 (1991), pp. 221–229.
- [203] Steven R Cummings, Michael C Nevitt and Sharon Kidd. “Forgetting falls”. In: *Journal of the American Geriatrics Society* 36.7 (1988), pp. 613–616.
- [204] Johannes Hilbe, Eva Schulc, Barbara Linder and Christa Them. “Development and alarm threshold evaluation of a side rail integrated sensor technology for the prevention of falls”. In: *International journal of medical informatics* 79.3 (2010), pp. 173–180.
- [205] Yueng Santiago Delahoz and Miguel Angel Labrador. “Survey on fall detection and fall prevention using wearable and external sensors”. In: *Sensors* 14.10 (2014), pp. 19806–19842.
- [206] Flo Wagner, Jenny Basran and Vanina Dal Bello-Haas. “A review of monitoring technology for use with older adults”. In: *Journal of Geriatric Physical Therapy* 35.1 (2012), pp. 28–34.
- [207] Caroline Rougier, Jean Meunier, Alain St-Arnaud and Jacqueline Rousseau. “Monocular 3D head tracking to detect falls of elderly people”. In: *Engineering in Medicine and Biology Society, 2006. EMBS’06. 28th Annual International Conference of the IEEE*. Vol. 2006. IEEE, New York City, USA. 2006, pp. 6384–6387.
- [208] Majd Alwan, Prabhu Jude Rajendran, Steve Kell, David Mack, Siddharth Dalal, Matt Wolfe and Robin Felder. “A smart and passive floor-vibration based fall detector for elderly”. In: *Information and Communication Technologies, 2006. ICTTA’06. 2nd*. Vol. 1. IEEE, Damascus, Syria. 2006, pp. 1003–1007.
- [209] Joanne Spetz, Joshua Jacobs and Carol Hatler. “Cost effectiveness of a medical vigilance system to reduce patient falls”. In: *Nursing Economics* 25.6 (2007), p. 333.
- [210] Bette Widder. “A new device to decrease falls”. In: *Geriatric Nursing* 6.5 (1985), pp. 287–288.
- [211] Robert G Cumming, Catherine Sherrington, Stephen R Lord, Judy M Simpson, Constance Vogler, Ian D Cameron and Vasi Naganathan. “Cluster randomised trial of a targeted multifactorial intervention to prevent falls among older people in hospital”. In: *BMJ* 336.7647 (2008), pp. 758–760.
- [212] Jiewen Zheng, Guang Zhang and Taihu Wu. “Design of automatic fall detector for elderly based on triaxial accelerometer”. In: *Bioinformatics and Biomedical Engineering, 2009. ICBBE 2009. 3rd International Conference*. IEEE, Beijing, China. 2009, pp. 1–4.

- [213] TK Sethuramalingam and A Vimalajuliet. “Design of MEMS based capacitive accelerometer”. In: *Mechanical and Electrical Technology (ICMET), 2010 2nd International Conference*. IEEE. 2010, pp. 565–568.
- [214] Samuel Ng Choon Po, Guo Dagang, Mohammad Dzulkifli Bin Mohyi Hapiipi, Nyan Myo Naing, Wei Jia Shen, Andojo Ongkodjojo and Francis Tay Eng Hock. “Overview of MEMS-Wear II-Incorporating MEMS Technology into smart shirt for Geriatric care”. In: *Journal of Physics: Conference Series*. Vol. 34. 1. IOP Publishing. 2006, p. 1079.
- [215] Francis EH Tay, D- G- Guo, L Xu, M- N- Nyan and K- L Yap. “MEMS-Wear-biomonitoring system for remote vital signs monitoring”. In: *Journal of the Franklin Institute* 346.6 (2009), pp. 531–542.
- [216] Urs Anliker, Jamie A Ward, Paul Lukowicz, Gerhard Troster, Francois Dolveck, Michel Baer, Fatou Keita, Eran B Schenker, Fabrizio Catarsi, Luca Coluccini et al. “AMON: a wearable multiparameter medical monitoring and alert system”. In: *IEEE Transactions on Information Technology in Biomedicine* 8.4 (2004), pp. 415–427.
- [217] PS Pandian, K Mohanavelu, KP Safeer, TM Kotresh, DT Shakunthala, Parvati Gopal and VC Padaki. “Smart Vest: Wearable multi-parameter remote physiological monitoring system”. In: *Medical engineering & Physics* 30.4 (2008), pp. 466–477.
- [218] Ramesh Rajagopalan, Irene Litvan and Tzyy-Ping Jung. “Fall prediction and prevention systems: recent trends, challenges, and future research directions”. In: *Sensors* 17.11 (2017), p. 2509.
- [219] Javad Razjouyan, Gurtej Singh Grewal, Cindy Rishel, Sairam Parthasarathy, Jane Mohler and Bijan Najafi. “Activity monitoring and heart rate variability as indicators of fall risk: proof-of-concept for application of wearable sensors in the acute care setting”. In: *Journal of Gerontological Nursing* 43.7 (2017), pp. 53–62.
- [220] Simon Freilich, Robert Barker et al. “Predicting falls risk in patients: The value of cardiovascular variability assessment”. In: *British Journal of Medical Practitioners* 2.4 (2009), pp. 44–48.
- [221] Md Shahiduzzaman. “Fall Detection by Accelerometer and Heart Rate Variability Measurement”. In: *Global Journal of Computer Science and Technology* 15.3 (2016), pp. 121–128.
- [222] R Nocua, N Noury, C Gehin, A Dittmar and E McAdams. “A new approach to improve the fall detection in elderly: monitoring of the autonomic nervous system activation”. In: *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany*. Springer, Berlin, Heidelberg. 2009, pp. 681–684.
- [223] Alexander M Chan, Nandakumar Selvaraj, Nima Ferdosi and Ravi Narasimhan. “Wireless patch sensor for remote monitoring of heart rate, respiration, activity, and falls”. In: *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*. IEEE. 2013, pp. 6115–6118.
- [224] M Isik, M Cankurtaran, BB Yavuz, A Deniz, B Yavuz, M Halil, Z Ulger, K Aytemir and S Arıođul. “Blunted baroreflex sensitivity: An underestimated cause of falls in the elderly?”. In: *European Geriatric Medicine* 3.1 (2012), pp. 9–13.

- [225] RJ Temple. “A regulatory authority’s opinion about surrogate endpoints”. In: *Clinical Measurement in Drug Evaluation* (1995), pp. 1–22.
- [226] Victor G De Gruttola, Pamela Clax, David L DeMets, Gregory J Downing, Susan S Ellenberg, Lawrence Friedman, Mitchell H Gail, Ross Prentice, Janet Wittes and Scott L Zeger. “Considerations in the evaluation of surrogate endpoints in clinical trials: summary of a National Institutes of Health workshop”. In: *Controlled Clinical Trials* 22.5 (2001), pp. 485–502.
- [227] Mavuto M Mukaka. “A guide to appropriate use of correlation coefficient in medical research”. In: *Malawi Medical Journal* 24.3 (2012), pp. 69–71.
- [228] Z Li, AA Chines and MP Meredith. “Statistical validation of surrogate endpoints: is bone density a valid surrogate for fracture?”. In: *Journal of Musculoskeletal and Neuronal Interactions* 4.1 (2004), p. 64.
- [229] PF Watson and A Petrie. “Method agreement analysis: a review of correct methodology”. In: *Theriogenology* 73.9 (2010), pp. 1167–1179.
- [230] J Martin Bland and Douglas G Altman. “Measuring agreement in method comparison studies”. In: *Statistical Methods in Medical research* 8.2 (1999), pp. 135–160.
- [231] Guillermo Macbeth, Eugenia Razumiejczyk and Rubén Daniel Ledesma. “Cliff’s Delta Calculator: A non-parametric effect size program for two groups of observations”. In: *Universitas Psychologica* 10.2 (2011), pp. 545–555.
- [232] IBM Corp. “IBM SPSS statistics for windows, version 22.0”. In: *Armonk, NY: IBM Corp* (2013).
- [233] Donna L Hudson and Maurice E Cohen. *Neural networks and artificial intelligence for biomedical engineering*. Wiley Online Library, 2000.
- [234] Nidhi H Ruparel, Nitin M Shahane and Devyani P Bhamare. “Learning from small data set to build classification model: A survey”. In: *Proc. IJCA Int. Conf. Recent Trends Eng. Technol.(ICRTET)*. 2013, pp. 23–26.
- [235] Partha Niyogi, Federico Girosi and Tomaso Poggio. “Incorporating prior information in machine learning by creating virtual examples”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2196–2209.
- [236] Torgyn Shaikhina, Dave Lowe, Sunil Daga, David Briggs, Robert Higgins and Natasha Khovanova. “Machine learning for predictive modelling based on small data in biomedical engineering”. In: *IFAC-PapersOnLine* 48.20 (2015), pp. 469–474.
- [237] Bradley Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. SIAM, Philadelphia, USA, 1982.
- [238] Ron Kohavi et al. “A study of cross-validation and bootstrap for accuracy estimation and model selection”. In: *IJCAI*. Vol. 14. 2. Stanford, CA. 1995, pp. 1137–1145.
- [239] Pawel Smialowski, Dmitrij Frishman and Stefan Kramer. “Pitfalls of supervised feature selection”. In: *Bioinformatics* 26.3 (2009), pp. 440–443.
- [240] Sasan Karamizadeh, Shahidan M Abdullah, Azizah A Manaf, Mazdak Zamani and Alireza Hooman. “An overview of principal component analysis”. In: *Journal of Signal and Information Processing* 4.03 (2013), p. 173.

- [241] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann and Ian H Witten. “The WEKA data mining software: an update”. In: *ACM SIGKDD Explorations Newsletter* 11.1 (2009), pp. 10–18.
- [242] Rashmi Dubey, Jiayu Zhou, Yalin Wang, Paul M Thompson, Jieping Ye, Alzheimer’s Disease Neuroimaging Initiative et al. “Analysis of sampling techniques for imbalanced data: an n=648 ADNI study”. In: *NeuroImage* 87 (2014), pp. 220–241.
- [243] M Mostafizur Rahman and DN Davis. “Addressing the class imbalance problem in medical datasets”. In: *International Journal of Machine Learning and Computing* 3.2 (2013), p. 224.
- [244] George Forman and Ira Cohen. “Learning from little: Comparison of classifiers given little training”. In: *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, Berlin, Heidelberg. 2004, pp. 161–172.
- [245] Nathalie Japkowicz. “The class imbalance problem: Significance and strategies”. In: *Proc. of the Intl Conf. on Artificial Intelligence*. 2000.
- [246] Miroslav Kubat, Stan Matwin et al. “Addressing the curse of imbalanced training sets: one-sided selection”. In: *ICML*. Vol. 97. Nashville, USA. 1997, pp. 179–186.
- [247] Show-Jane Yen and Yue-Shi Lee. “Cluster-based sampling approaches to imbalanced data distributions”. In: *International Conference on Data Warehousing and Knowledge Discovery*. Springer, Berlin, Heidelberg. 2006, pp. 427–436.
- [248] Jun Liu, Jianhui Chen and Jieping Ye. “Large-scale sparse logistic regression”. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2009, pp. 547–556.
- [249] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall and W Philip Kegelmeyer. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial Intelligence Research* 16 (2002), pp. 321–357.
- [250] T Maruthi Padmaja, P Radha Krishna and Raju S Bapi. “Majority filter-based minority prediction (MFMP): An approach for unbalanced datasets”. In: *TENCON 2008-2008 IEEE Region 10 Conference*. IEEE, Piscataway, N.J. 2008, pp. 1–6.
- [251] Andrew Estabrooks, Taeho Jo and Nathalie Japkowicz. “A multiple resampling method for learning from imbalanced data sets”. In: *Computational Intelligence* 20.1 (2004), pp. 18–36.
- [252] Nitesh V Chawla, Nathalie Japkowicz and Aleksander Kotcz. “Special issue on learning from imbalanced data sets”. In: *ACM Sigkdd Explorations Newsletter* 6.1 (2004), pp. 1–6.
- [253] Foster Provost and Tom Fawcett. “Robust classification for imprecise environments”. In: *Machine learning* 42.3 (2001), pp. 203–231.
- [254] Nozomi Sato, Shinji Miyake, Jun’ichi Akatsu and Masaharu Kumashiro. “Power spectral analysis of heart rate variability in healthy young women during the normal menstrual cycle”. In: *Psychosomatic Medicine* 57.4 (1995), pp. 331–335.
- [255] AL Goldberger, LAN Amaral, L Glass, JM Hausdorff, P Ch Ivanov, RG Mark, JE Mietus, GB Moody, CK Peng and HE Stanley. “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals.” In: *Circulation* 101.23 (2000), e215–e220.

- [256] W Zong, GB Moody and D Jiang. “A robust open-source algorithm to detect onset and duration of QRS complexes”. In: *Computers in Cardiology, 2003*. IEEE. 2003, pp. 737–740.
- [257] George B Moody. “Evaluating ECG analyzers”. In: *WFDB Applications Guide* (2003).
- [258] MH Asyali. “Discrimination power of long-term heart rate variability measures”. In: *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*. Vol. 1. IEEE. 2003, pp. 200–203.
- [259] Georg Thimm and Emile Fiesler. *Optimal setting of weights, learning rate, and gain*. Tech. rep. IDIAP, 1997.
- [260] Xiaoyuan Zhang, Daoyin Qiu and Fuan Chen. “Support vector machine with parameter optimization by a novel hybrid method and its application to fault diagnosis”. In: *Neurocomputing* 149 (2015), pp. 641–651.
- [261] LV Rajani Kumari, Y Padma Sai and N Balaji. “SVM to Classify Mental Stress in Drivers Using HRV Analysis”. In: *Automation and Autonomous System* 9.9 (2017), pp. 181–185.
- [262] Anil K Ghosh. “On optimum choice of k in nearest neighbor classification”. In: *Computational Statistics & Data Analysis* 50.11 (2006), pp. 3113–3123.
- [263] Diego Raphael Amancio, Cesar Henrique Comin, Dalcimar Casanova, Gonzalo Travieso, Odemir Martinez Bruno, Francisco Aparecido Rodrigues and Luciano da Fontoura Costa. “A systematic comparison of supervised classifiers”. In: *PloS One* 9.4 (2014), e94137.
- [264] VK Chadha. “Sample size determination in health studies”. In: *NTI Bulletin* 42.3&4 (2006), pp. 55–62.
- [265] Kevin K Dobbin and Richard M Simon. “Optimally splitting cases for training and testing high dimensional classifiers”. In: *BMC Medical Genomics* 4.1 (2011), p. 31.
- [266] Michael Reardon and Marek Malik. “Changes in heart rate variability with age”. In: *Pacing and Clinical Electrophysiology* 19.11 (1996), pp. 1863–1866.
- [267] Phyllis K Stein, Robert E Kleiger and Jeffrey N Rottman. “Differing effects of age on heart rate variability in men and women”. In: *The American Journal of Cardiology* 80.3 (1997), pp. 302–305.
- [268] Jagmeet P Singh, Martin G Larson, Hisako Tsuji, Jane C Evans, Christopher J ODonnell and Daniel Levy. “Reduced heart rate variability and new-onset hypertension”. In: *Hypertension* 32.2 (1998), pp. 293–297.
- [269] Emily B Schroeder, Duanping Liao, Lloyd E Chambless, Ronald J Prineas, Gregory W Evans and Gerardo Heiss. “Hypertension, blood pressure, and heart rate variability”. In: *Hypertension* 42.6 (2003), pp. 1106–1111.
- [270] Barathi S Subramaniam. “Influence of Body Mass Index on Heart Rate Variability (HRV) in evaluating cardiac function in adolescents of a selected Indian population”. In: *Italian Journal of Public Health* 8.2 (2012), pp. 149–155.
- [271] Ivana Antelmi, Rogério Silva De Paula, Alexandre R Shinzato, Clóvis Araújo Peres, Alfredo José Mansur and Cesar José Grupi. “Influence of age, gender, body mass index, and functional capacity on heart rate variability in a cohort of subjects without heart disease”. In: *The American Journal of Cardiology* 93.3 (2004), pp. 381–385.

- [272] Robert E Kleiger, Phyllis K Stein and J Thomas Bigger. “Heart rate variability: measurement and clinical utility”. In: *Annals of Noninvasive Electrocardiology* 10.1 (2005), pp. 88–101.
- [273] Jon T Ingjaldsson, Jon C Laberg and Julian F Thayer. “Reduced heart rate variability in chronic alcohol abuse: relationship with negative mood, chronic thought suppression, and compulsive drinking”. In: *Biological Psychiatry* 54.12 (2003), pp. 1427–1436.
- [274] Susan Folkman, Richard S Lazarus, Christine Dunkel-Schetter, Anita DeLongis and Rand J Gruen. “Dynamics of a stressful encounter: cognitive appraisal, coping, and encounter outcomes.” In: *Journal of personality and social psychology* 50.5 (1986), p. 992.
- [275] Andrew Steptoe and Claus Vogele. “Methodology of mental stress testing in cardiovascular research”. In: *Circulation* 83.4 Suppl (1991), pp. II14–II24.
- [276] Yoshikawa Hoshikawa and YOSHIHARU Yamamoto. “Effects of Stroop color-word conflict test on the autonomic nervous system responses”. In: *American journal of physiology-Heart and circulatory physiology* 272.3 (1997), H1113–H1121.
- [277] Sylvie Hébert, Renée Béland, Odrée Dionne-Fournelle, Martine Crête and Sonia J Lupien. “Physiological stress response to video-game playing: the contribution of built-in music”. In: *Life sciences* 76.20 (2005), pp. 2371–2380.
- [278] W Miles Cox, Javad Salehi Fadardi and Emmanuel M Pothos. “The addiction-stroop test: Theoretical considerations and procedural recommendations.” In: *Psychological bulletin* 132.3 (2006), p. 443.
- [279] Charles J Golden. “A group version of the Stroop Color and Word Test”. In: *Journal of Personality Assessment* 39.4 (1975), pp. 386–388.
- [280] Clj Bench, CD Frith, PM Grasby, KJ Friston, E Paulesu, RSJ Frackowiak and RJ Dolan. “Investigations of the functional anatomy of attention using the Stroop test”. In: *Neuropsychologia* 31.9 (1993), pp. 907–922.
- [281] Peter E Comalli Jr, Seymour Wapner and Heinz Werner. “Interference effects of Stroop color-word test in childhood, adulthood, and aging”. In: *The Journal of genetic psychology* 100.1 (1962), pp. 47–53.
- [282] J Ridley Stroop. “Studies of interference in serial verbal reactions.” In: *Journal of Experimental Psychology* 18.6 (1935), p. 643.
- [283] Sandro Rubichi and Roberto Nicoletti. “The Simon effect and handedness: Evidence for a dominant-hand attentional bias in spatial coding”. In: *Perception & Psychophysics* 68.7 (2006), pp. 1059–1069.
- [284] Frederiek F Van Doormaal, Gary E Raskob, Bruce L Davidson, Hervé Decousus, Alexander Gallus, AW Lensing, Franco Piovella, Martin H Prins, Harry R Büller et al. “Treatment of venous thromboembolism in patients with cancer: subgroup analysis of the Matisse clinical trials”. In: *Thromb Haemost* 101.4 (2009), pp. 762–769.
- [285] Robert Coe. “Its the Effect Size, Stupid”. In: *The British Educational Research Association Annual Conference of the British Educational Research Association, University of Exeter, England*. Vol. 12. 2002, p. 14.

- [286] JPA Delaney and DA Brodie. “Effects of short-term psychological stress on the time and frequency domains of heart-rate variability”. In: *Perceptual and motor skills* 91.2 (2000), pp. 515–524.
- [287] Alberto Malliani, Federico Lombardi and Massimo Pagani. “Power spectrum analysis of heart rate variability: a tool to explore neural regulatory mechanisms.” In: *British Heart Journal* 71.1 (1994), p. 1.
- [288] Frank A Treiber, Thomas Kamarck, Neil Schneiderman, David Sheffield, Gaston Kapuku and Teletia Taylor. “Cardiovascular reactivity and development of preclinical and clinical disease states”. In: *Psychosomatic medicine* 65.1 (2003), pp. 46–62.
- [289] Authors/Task Force Members, Giuseppe Mancia, Robert Fagard, Krzysztof Narkiewicz, Josep Redon, Alberto Zanchetti, Michael Böhm, Thierry Christiaens, Renata Cifkova, Guy De Backer et al. “2013 ESH/ESC guidelines for the management of arterial hypertension: the Task Force for the Management of Arterial Hypertension of the European Society of Hypertension (ESH) and of the European Society of Cardiology (ESC)”. In: *European Heart Journal* 34.28 (2013), pp. 2159–2219.
- [290] Laurence Z Rubenstein. “Falls in older people: epidemiology, risk factors and strategies for prevention”. In: *Age and ageing* 35.suppl.2 (2006), pp. ii37–ii41.
- [291] Leandro Pecchia, Paolo Melillo, Mario Sansone and Marcello Bracale. “Discrimination power of short-term heart rate variability measures for CHF assessment”. In: *IEEE Transactions on Information Technology in Biomedicine* 15.1 (2011), pp. 40–46.
- [292] Leandro Pecchia, Paolo Melillo and Marcello Bracale. “Remote health monitoring of heart failure with data mining via CART method on HRV features”. In: *IEEE Transactions on Biomedical Engineering* 58.3 (2011), pp. 800–804.
- [293] Mirza Mansoor Baig, Hamid Gholamhosseini and Martin J Connolly. “A comprehensive survey of wearable and wireless ECG monitoring systems for older adults”. In: *Medical and Biological Engineering and Computing* 51.5 (2013), p. 485.
- [294] Behzad Mirmahboub, Shadrokh Samavi, Nader Karimi and Shahram Shirani. “Automatic monocular system for human fall detection based on variations in silhouette area”. In: *IEEE Transactions on Biomedical Engineering* 60.2 (2013), pp. 427–436.
- [295] Yun Li, KC Ho and Mihail Popescu. “Efficient source separation algorithms for acoustic fall detection using a Microsoft Kinect”. In: *IEEE Transactions on Biomedical Engineering* 61.3 (2014), pp. 745–755.
- [296] Erik E Stone and Marjorie Skubic. “Fall detection in homes of older adults using the Microsoft Kinect”. In: *IEEE journal of biomedical and health informatics* 19.1 (2015), pp. 290–301.
- [297] Juan Cheng, Xiang Chen and Minfen Shen. “A framework for daily activity monitoring and fall detection based on surface electromyography and accelerometer signals”. In: *IEEE journal of biomedical and health informatics* 17.1 (2013), pp. 38–45.
- [298] Xin Ma, Haibo Wang, Bingxia Xue, Mingang Zhou, Bing Ji and Yibin Li. “Depth-based human fall detection via shape features and improved extreme learning machine”. In: *IEEE journal of biomedical and health informatics* 18.6 (2014), pp. 1915–1922.

- [299] Paolo Melillo, Ada Orrico, Franco Chirico, Leandro Pecchia, Settimio Rossi, Francesco Testa and Francesca Simonelli. “Identifying fallers among ophthalmic patients using classification tree methodology”. In: *PLoS one* 12.3 (2017), e0174083.
- [300] Saman Parvaneh, Bijan Najafi, Nima Toosizadeh, Irbaz Bin Riaz and Jane Mohler. “Is there any association between ventricular ectopy and falls in community-dwelling older adults?” In: *Computing in Cardiology Conference (CinC), 2016*. IEEE, Vancouver, BC. 2016, pp. 433–436.

Appendices

Appendix A

Matlab Tools

A.1 Matlab tool for meta-analysis

The Matlab tool for meta-analysis is reported in this section.

```
1 function Call_MetaAnalizza(Matrix)
2 %%This MATLAB tool was created to run meta-analysis, using means and
3 %%standard deviations from different studies exploring two ...
4 %%different groups
5 %%(i.e., treatments and controls groups).
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 %% Input: Matrix=[ ID #_A Mean_A SD_A #_B Mean_B SD_B]
8 %%ID:Study identification
9 %%#_A: number of subjects in control group
10 %%#_B: number of subjects in treatment group
11 %%Mean_A and Mean_B: mean values for control and treatment groups
12 %%respectively
13 %%SD_A and SD_B: standard deviation value for control and ...
14 %%treatment groups
15 %%respectively
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17 %% Output: StatFinal and Forest Plot
18 %%StatFinal: one numerical row reporting: the total population, ...
19 %%heterogeneity
20 %%and related p-value, model (i.e., random or fixed), the effect ...
21 %%size, 95% CI and p-value.
22 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23 %%Created by Rossana Castaldo, Univeristy of Warwick, Feb 2014.
24 %%Revised by Rossana Castaldo, Dicember 2016
25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26 %% Studies' IDs
```

```

23     texts=table2array(Matrix(:,1));
24     Matrix=table2array(Matrix(:,2:end));
25     % This Function will choose if a Random or Fixed Model needs to ...
        be used
26     [Model Mod_num p Q Isq]=ModelSelection(Matrix);
27     %This function will compute the weight for each study based on random
28     %or fixed model
29     [T s var_T Low High w PercentageWeightofStudy ...
        tprob]=WeightCalculation(Matrix, Mod_num)
30     % Final result of the meta-analysis
31     [MD var_MD SD_MD LowS HighS Sum_sub_total pS]=MainStat(w,T, Matrix)
32     StatFinal=table(p, Q, Isq, Model, MD, LowS, HighS, pS);
33     %This function will compute the statistical analysis to obtain a ...
        Forest
34     %plot
35     texts=[texts; 'Polled']
36
37     T=[T; MD];
38     Low=[ Low; LowS];
39     High=[ High; HighS]
40     N=[ Matrix(:,1);sum(Matrix(:,1))];
41     p=[ tprob pS ];
42     celldata={texts, T, Low, High,N, p};
43     figure, Forest (celldata,PercentageWeightofStudy)
44
45     end

```

```

1 function [Model Mod_num p Q Isq]=ModelSelection(Matrix)
2
3     [p Q,Isq]=Q_test(Matrix);
4     if p>0.05
5         Model=string({'Fixed'});
6         Mod_num=1;
7     else
8
9         Model=string({'Random'});
10        Mod_num=2;
11    end
12 end

```

```

1 function [p Q Isq]=Q_test(Matrix)
2     [T s var_T Low High w]=WeightCalculation(Matrix,1); %here we use ...
        fixed model to calculate weight, therefore,Mod_num=1

```

```

3     [n_study 1]=size(Matrix);
4     Q=w'*(T-(w'*T)/sum(w)).^2;
5     p = chi2cdf(Q,n_study-1,'upper');
6     Isq=100*((Q-(n_study-1))/Q);
7     end

1 function [T s var_T Low High w PercentageWeightofStudy ...
           tprob]=WeightCalculation(Matrix, Mod_num)
2 %MATRICE=[#_A Mean_A SD_A #_B Mean_B SD_B]
3 %OUT=B-A
4 [n_study 1]=size(Matrix);
5 if (Mod_num==1); %Fixed Model
6     n_A=Matrix(:,1);
7     Mean_A=Matrix(:,2);
8     SD_A=Matrix(:,3);
9
10    n_B=Matrix(:,4);
11    Mean_B=Matrix(:,5);
12    SD_B=Matrix(:,6);
13
14    s=((n_A-1).*SD_A.^2+(n_B-1).*SD_B.^2)./(n_A+n_B-2).^0.5;
15    var_T=s.^2.*(1./n_A+1./n_B);
16
17    T=Mean_B-Mean_A;
18    Low= T-1.96*var_T.^0.5;
19    High=T+1.96*var_T.^0.5;
20    CI_95=[T-1.96*var_T.^0.5 T+1.96*var_T.^0.5];
21    w=1./var_T;
22    PercentageWeightofStudy= 100.*(w./sum(w));
23
24    for i=1:n_study
25        s(i)=(((n_A(i)-1).*SD_A(i).^2+(n_B(i)-1).*SD_B(i).^2)./...
26            (n_A(i)+n_B(i)-2)).^0.5;
27        var_Tind(i)=s(i).^2.*(1./n_A(i)+1./n_B(i));
28        Tind(i)=Mean_B(i)-Mean_A(i);
29        Low(i)=Tind(i)-1.96*var_Tind(i).^0.5
30        High(i)= Tind(i)+1.96*var_Tind(i).^0.5;
31        % Calculate T-Statistic
32        v(i)= (n_A(i)+n_B(i))-2;
33        tval(i) = ( Mean_B(i)-Mean_A(i)) / ...
34            sqrt((SD_B(i)^2/n_B(i))+((SD_A(i)^2)/n_A(i)));
35        tdist2T = @(t,v) (1-betainc(v/(v+t^2),v/2,0.5)); % ...
36            2-tailed t-distribution
37        tdist1T = @(t,v) 1-(1-tdist2T(t,v))/2; % ...
38            1-tailed t-distribution

```

```

36         tprob(i) = 1-[tdist2T(tval(i),v(i))]
37     end
38
39
40
41 else %Random Model
42     [n_study 1]=size(Matrix);
43     n_A=Matrix(:,1);
44     Mean_A=Matrix(:,2);
45     SD_A=Matrix(:,3);
46
47     n_B=Matrix(:,4);
48     Mean_B=Matrix(:,5);
49     SD_B=Matrix(:,6);
50
51     s=((n_A-1).*SD_A.^2+(n_B-1).*SD_B.^2)./(n_A+n_B-2).^0.5;
52     var_T=s.^2.*(1./n_A+1./n_B);
53     T=Mean_B-Mean_A;
54     Low=T-1.96*var_T.^0.5;
55     High=T+1.96*var_T.^0.5;
56     CI_95=[T-1.96*var_T.^0.5 T+1.96*var_T.^0.5];
57
58     w1=1./var_T;
59
60     w_mean=sum(w1)/n_study;
61
62     U=sum(w1)-(sum(w1.^2)/sum(w1));
63     %Check
64     [p Q]=Q_test(Matrix);
65
66     if (Q<=(n_study-1))
67         tau_sq=0
68     else
69         tau_sq=(Q-(n_study-1))/U;
70     end
71
72     w=(1./((1./w1)+tau_sq));
73     PercentageWeightofStudy= 100.*(w./sum(w));
74
75     for i=1:n_study
76         s(i)=(((n_A(i)-1).*SD_A(i).^2+(n_B(i)-1).*SD_B(i).^2)./...
77             (n_A(i)+n_B(i)-2)).^0.5;
78         var_Tind(i)=s(i).^2.*(1./n_A(i)+1./n_B(i));
79         Tind(i)=Mean_B(i)-Mean_A(i);
80         Low(i)=Tind(i)-1.96*var_Tind(i).^0.5
81         High(i)= Tind(i)+1.96*var_Tind(i).^0.5;

```

```

82     % Calculate T-Statistic
83     v(i)= (n_A(i)+n_B(i))-2;
84     tval(i) = ( Mean_B(i)-Mean_A(i)) / ...
            sqrt((SD_B(i)^2/n_B(i))+((SD_A(i)^2)/n_A(i)));
85     tdist2T = @(t,v) (1-betainc(v/(v+t^2),v/2,0.5)); % ...
            2-tailed t-distribution
86     tdist1T = @(t,v) 1-(1-tdist2T(t,v))/2; % ...
            1-tailed t-distribution
87     tprob(i) = 1-[tdist2T(tval(i),v(i))]
88     end
89 end

1 function [MD var_MD SD_MD Low High Sum_sub_total p]=MainStat(w,T, ...
    Matrix)
2 n_A=Matrix(:,1);
3 n_B=Matrix(:,4);
4 Sum_sub_total=sum(Matrix(:,1))
5 MD=(w'*T/sum(w));
6 var_MD=(1/sum(w));
7 SD_MD=var_MD^.5;
8 Low=MD-1.96*var_MD^.5
9 High=MD+1.96*var_MD^.5;
10 CI_95=[Low High];
11 % Calculate Z-Statistic
12 Z=MD/SD_MD;
13 p=normcdf(Z);
14 Summation=[MD, Low, High, Sum_sub_total, p];
15 end

1 function Forest(cellldata, PercentageWeightofStudy) %texts, ...
    mean,low,high,size2,p
2 texts=cellldata{1,1}
3 texts=flip(texts')
4 mean=cellldata{1,2}
5 mean=flip(mean')
6 low=cellldata{1,3}
7 low=flip (low)
8 high=cellldata{1,4}
9 high=flip(high)
10 size2=cellldata{1,5}
11 size2=flip(size2')
12 p=cellldata{1,6};
13 p=flip(p');

```

```

14 PercentageWeightofStudy=flip(PercentageWeightofStudy);
15 subplot(1,2,1)
16 n=length(texts);
17
18 for n=1:length(texts)
19     text(0,n,texts{n});
20     numbers = sprintf('%9.3f %9.3f %9.3f %9.3f', ...
        mean(n),low(n),high(n), p(n));
21     text(0.5,n,numbers);
22 end
23 text(0,n+1,'Study')
24 text(0.5,n+1,'MD')
25 text(0.65,n+1,'Low')
26 text(0.8,n+1,'High')
27 text(0.95,n+1,'p-val');
28
29 axis([0,1.5,0,length(texts)+1])
30 set(gca,'visible','off')
31
32 subplot(1,2,2)
33 plot([low high]',[1;1]*(1:length(mean)),'k')
34 axis([-max(abs(low))-1,max(abs(high))+1,0,length(mean)+1])
35 hold on
36 if sign(low(1))==sign(high(1)) && sign(low(1))==-1
37     sizeSumm=-(abs(high(1))+abs(low(1)))/2
38 else
39     if sign(low(1))==sign(high(1))&& sign(low(1))==1
40         sizeSumm=(abs(high(1))+abs(low(1)))/2;
41     else
42 if sign(low(1))~ sign(high(1))
43 sizeSumm=(high(1)+low(1))/2;
44 end
45     end
46 end
47 for i=length(texts):-1:2
48
49     plot(mean(i),i,'ks','markerSize',...
50         PercentageWeightofStudy(i-1)+2,...
51         'MarkerEdgeColor','k','MarkerFaceColor','k')
52     set(gcf, 'Units', 'Normalized', 'OuterPosition', ...
53         [0, 0.04, 1, 0.96]);
54
55 end
56 hold on %
57 %
58 x=[sizeSumm high(1) sizeSumm low(1)];

```

```

59 y=[0.8 1 1.2 1];
60 plot(x,y)
61 fill(x,y,'k')
62 hold on
63 plot([0 0], ylim,'--k') % line no effect % because mean difference
64 xlabel('Mean Difference')

```

A.2 Matlab tool to identify biomedical surrogates

The Matlab tool to identify surrogates is reported in this section.

```

1 function[results]=main ()
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This Matlab tool was created to identify surrogate features. It
4 %%differentiates whether the features need to be investigate in one
5 %%condition (i.e., resting) or two conditions (i.e., resting and stress)
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 %%The final output: A table containing the main descriptive ...
8     statistics and
9     %%a table with the final surrogate features
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 %%Created by Rossana Castaldo, Univeristy of Warwick, 2017
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 %%USER input:
14 prompt = 'Do you have one condition? Enter yes if you do or no if you ...
15     have two conditions: ';
16 x = input(prompt,'s');
17 switch x
18     case 'yes'
19         %call for function using one condition
20         [Tablestat,SurrogateFeature] = Surrugateonecondition(x);
21     case 'no'
22         %call for function using two conditions
23         [Tablestat,TableSTATSurrogates,SurrogateFeature] = ...
24             Surrugatetwoconditions(x); %call for function using two ...
25             conditions
26     otherwise
27         display('error! Please enter yes or no')
28 end
29 results={Tablestat, SurrogateFeature};
30 end

```

```

1
2 function [Tablestat,SurrogateFeatureMatrices] = Surrugateonecondition(x)
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 %%This tool was developed to indentify the subset of surrogates ...
   features comparing the
5 %%benchmark features (normal length) and ultra-short
6 %%features (shorter length). In particular, this script was used to ...
   compare
7 %%only two different time scales, but it can be adapted if more time ...
   scales
8 %%are available.
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 %%USER input: files .csv with features in columns and as header the
11 %%features' names
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 %%The output of this tool produces basic stat of the datasets
14 %%(using parametric or non-parametric methods) and the subset of features
15 %%that are identified as good surrogates of the benchmark.
16 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
17 display('Select data set benchmark...')
18 [filename, pathname] = uigetfile('*.csv', ' Please select the input ...
   file');
19 complete_path = strcat(pathname, filename);
20 benchmarktbl = readtable(complete_path);
21 VarNames=benchmarktbl.Properties.VariableNames;
22 benchmarkarray=table2array(benchmarktbl);
23
24 display('Select data set 1st time length...')
25 [filename, pathname] = uigetfile('*.csv', ' Please select the input ...
   file');
26 complete_path = strcat(pathname, filename);
27 FirstTimeLengthtbl = readtable(complete_path);
28 FirstTimeLengtharray=table2array(FirstTimeLengthtbl);
29 %%
30 s=size(benchmarkarray);
31 if ~isequal(s,size(FirstTimeLengtharray));
32     error('data1 and data2 must have the same size');
33 end
34 %%
35 [Rows Columns]=size(benchmarkarray);
36 %test for normality
37 for i=1:Columns
38 [h(i),p(i)] = lillietest(benchmarkarray(:,i)); %if h=1 the feature is ...
   non-normally distributed
39 end
40 CountNONNormaly=sum(h==1);

```

```

41 if CountNONNormaly==Columns
42     display('All features are non-normally distributed')
43 else
44     if CountNONNormaly==0
45         display('Data is normally distributed')
46     else
47         if CountNONNormaly>(Columns/2)
48             display('Many features are non-normally distributed, ...
                strongly reccomanded to use non-parametric test')
49         else
50             if CountNONNormaly<(Columns/2)
51                 display('Some features are non-normally distributed, ...
                    strongly reccomanded to use non-parametric test or ...
                    apply log-transformation')
52             end
53         end
54     end
55 end
56 Condition=(CountNONNormaly≠0);
57 if Condition
58     prompt = 'Do you want to log-transform your data? Enter yes if you do ...
                or no if you do not: ';
59     str = input(prompt,'s');
60     switch str
61         case 'yes'
62             for i=1:Columns
63                 benchmarkarray(:,i)=log( benchmarkarray(:,i));
64                 FirstTimeLengtharray(:,i)=log(FirstTimeLengtharray(:,i));
65             end
66             CountNONNormaly=0;
67             Condition=(CountNONNormaly==0);
68         case 'no'
69             CountNONNormaly≠0;
70             Condition=(CountNONNormaly==0);
71         otherwise
72     end
73 end
74 %%
75 %General statistic indexes
76 TablestatBenchmark=Stat(benchmarkarray,VarNames',Condition, x)
77 TablestatFirstTime=Stat(FirstTimeLengtharray,VarNames',Condition, x)
78 Tablestat={TablestatBenchmark, TablestatFirstTime};
79 %%
80 % First step: correlation analysis
81 [D]=correlation(benchmarkarray, FirstTimeLengtharray, Condition, x)
82 rho_M=logical(D.diagMask)';

```

```

83 VarNamesCorrelated=VarNames(rho_M);
84 %%
85 %Second step is visual inspection using parametric Bland-Altman Plot
86 [cr, fig, statsStruct]= ...
    BlandAltmanPlts(benchmarkarray,FirstTimeLengtharray,Condition, x);
87
88 l=sum(rho_M==1);
89 rho_M=logical(rho_M);
90 %Selection of the features that are significant correlated
91 CorrelatedDataFirstTimeLengtharray=FirstTimeLengtharray(:,rho_M);
92 CorrelatedDatabenchmarkarray=benchmarkarray(:,rho_M);
93 %%
94 %Third step is measuring the effect size
95 [D,SurrogateFeatureMatrices]=StatOneCond(CorrelatedDatabenchmarkarray, ...
    CorrelatedDataFirstTimeLengtharray,VarNamesCorrelated, Condition);
96
97 end

1 function [Tablestat,TableSTATSurrogates,SurrogateFeatureMatrices] = ...
    Surrugatetwoconditions(x)
2 %%This tool was developed to indentify the subset of surrogates ...
    features comparing
3 %%the benchmark features (normal length) and ultra-short
4 %%features (shorter length)in two different conditions. In ...
    particular, this script was used to compare
5 %%only two different time scales with the benchmark, but it can be ...
    adapted if more time scales
6 %%are available. The outputs of this tool are: basic stat of the ...
    dataset at
7 %%different time scale (using parametric or non-parametric methods) ...
    and the subset of features
8 %%that are identified as good surrogates of the benchmark at ...
    different time scales.
9 %%
10 %Input Data Section
11
12 display('Select data set benchmark...')
13 [filename, pathname] = uigetfile('*.csv', ' Please select the input ...
    file');
14 complete_path = strcat(pathname, filename);
15 benchmarktbl = readtable(complete_path);
16 VarNames=benchmarktbl.Properties.VariableNames;
17 VarNames=VarNames(1:end-1);
18 benchmarkarray=table2array(benchmarktbl);
19

```

```

20 display('Select data set 1st time length...')
21 [filename, pathname] = uigetfile('*.csv', ' Please select the input ...
    file');
22 complete_path = strcat(pathname, filename);
23 FirstTimeLengthtbl = readtable(complete_path);
24 FirstTimeLengtharray=table2array(FirstTimeLengthtbl);
25
26
27 display('Select data set 2nd time length...')
28 [filename, pathname] = uigetfile('*.csv', ' Please select the input ...
    file');
29 complete_path = strcat(pathname, filename);
30 SecondTimeLengthtbl = readtable(complete_path);
31 SecondTimeLengtharray=table2array(SecondTimeLengthtbl);
32 %%
33 s=size(benchmarkarray);
34 if ~isequal(s, size(FirstTimeLengtharray), size(SecondTimeLengtharray));
35     error('Matrices must have the same size');
36 end
37 [Rows Columns]=size(benchmarkarray);
38 %%
39 %Test for normality
40 for i=1:Columns
41 [h(i),p(i)] = lillietest(benchmarkarray(:,i)); %if h=1 the feature is ...
    non normally distributed
42 end
43 CountNONNormally=sum(h==1);
44 if CountNONNormally==Columns
45     display('All features are non-normally distributed')
46 else
47     if CountNONNormally==0
48         display('Data is normally distributed')
49     else
50         if CountNONNormally>(Columns/2)
51             display('Many features are non-normally distributed, ...
                strongly reccomanded to use non-parametric test')
52         else
53             if CountNONNormally<(Columns/2)
54                 display('Some features are non-normally distributed, ...
                    strongly reccomanded to use non-parametric test ...
                    or apply log-transformation')
55             end
56         end
57     end
58 end
59 %%

```

```

60 Condition=(CountNONNormally≠0);
61 if Condition
62 prompt = 'Do you want to log-transform your data? Enter yes if you do ...
           or no if you do not: ';
63 str = input(prompt,'s');
64 switch str
65     case 'yes'
66         for i=1:Columns
67             benchmarkarray(:,i)=log( benchmarkarray(:,i));
68             FirstTimeLengtharray(:,i)=log(FirstTimeLengtharray(:,i));
69             SecondTimeLengtharray(:,i)=log(SecondTimeLengtharray(:,i));
70         end
71         CountNONNormally=0;
72         Condition=(CountNONNormally==0);
73     case 'no'
74         CountNONNormally≠0;
75         Condition=(CountNONNormally==0);
76     otherwise
77 end
78 end
79 %%
80 Tablestat1=Stat(benchmarkarray, VarNames',Condition,x);
81 Tablestat2=Stat(FirstTimeLengtharray,VarNames', Condition,x);
82 Tablestat3=Stat(SecondTimeLengtharray, VarNames', Condition,x);
83 field1='TableStatBenchmark';
84 field2='TableStatFirstTimeLength';
85 field3='TableStatSecondTimeLength';
86 Tablestat=struct(field1,Tablestat1, field2, Tablestat2, field3, ...
                  Tablestat3);
87 %%
88 % The first step correlation analysis
89 D1=correlation(benchmarkarray,FirstTimeLengtharray, Condition,x);
90 D2=correlation(benchmarkarray,SecondTimeLengtharray, Condition,x);
91 %%
92 %The second step is visual inspection using parametric Bland-Altman Plot
93 [cr1, fig, statsStruct1]= ...
    BlandAltmanPlts(benchmarkarray,FirstTimeLengtharray, Condition);
94 [cr2, fig, statsStruct2]= ...
    BlandAltmanPlts(benchmarkarray,SecondTimeLengtharray, Condition);
95 %%
96 %% Decision Rule to select only highly significant correlated ...
    features among all the time scales investigated
97 for i=1:Columns-1
98     if D1.diagExp(i)==1 && D1.diagExp(i)==D1.diagControl(i)
99         rhoF1(i)=1;
100     else

```

```

101         rhoF1(i)=0;
102     end
103     if D2.diagControl(i)==1 && D2.diagControl(i)==D2.diagExp(i)
104         rhoF2(i)=1;
105     else
106         rhoF2(i)=0;
107     end
108 end
109
110
111
112 rhoF1=logical(rhoF1);%Indices of the features that are highly ...
        significant between the benchmark and the first time scale
113 rhoF2=logical(rhoF2); %Indices of the features that are highly ...
        significant between the benchmark and the Second time scale
114 CorrelatedVarNamesF1=VarNames(rhoF1');
115 CorrelatedVarNamesF2=VarNames(rhoF2');
116 CorrelatedDataFirstTimeLengtharray=FirstTimeLengtharray(:,rhoF1);
117 group2=FirstTimeLengtharray(:,end);
118 CorrelatedDataFirstTimeLengtharray=...
119     [CorrelatedDataFirstTimeLengtharray,group2];
120 CorrelatedDatabenchmarkarrayF1=benchmarkarray(:,rhoF1)
121 group1=benchmarkarray(:,end);
122 CorrelatedDatabenchmarkarrayF2=benchmarkarray(:,rhoF2)
123 CorrelatedDatabenchmarkarrayF1= ...
124     [CorrelatedDatabenchmarkarrayF1,group1];
125 CorrelatedDatabenchmarkarrayF2= ...
126     [CorrelatedDatabenchmarkarrayF2,group1];
127 CorrelatedDataSecondTimeLengtharray=SecondTimeLengtharray(:,rhoF2);
128 group3=SecondTimeLengtharray(:,end);
129 CorrelatedDataSecondTimeLengtharray=...
130     [CorrelatedDataSecondTimeLengtharray,group3];
131 %%
132 %The third step is measuring the effect via statistical test
133 [Tablestat1.reducedF1,PvaluesF1]=Stat(CorrelatedDatabenchmarkarrayF1, ...
134     ...
135     CorrelatedVarNamesF1', Condition,x);
136 [Tablestat1.reducedF2, ...
137     PvaluesF2]=Stat(CorrelatedDatabenchmarkarrayF2,...
138     CorrelatedVarNamesF2', Condition,x);
139 [Tablestat2.reduced, ...
140     PvaluesFirt]=Stat(CorrelatedDataFirstTimeLengtharray,...
141     CorrelatedVarNamesF1', Condition,x);
142 [Tablestat3.reduced, ...
143     PvaluesSecond]=Stat(CorrelatedDataSecondTimeLengtharray,...
144     CorrelatedVarNamesF2', Condition,x);

```

```

139
140 PosSignChanging1F1=(PvaluesF1<0.05);
141 %Var_NameF1=CorrelatedVarNames(PosSignChanging1F1) '
142 PosSignChanging1F2=(PvaluesF2<0.05);
143 %Var_NameF2=CorrelatedVarNames(PosSignChanging1F2) '
144 PosSignChanging2=PvaluesFirt<0.05;
145 Var_Name2=CorrelatedVarNamesF1(PosSignChanging2) '
146 PosSignChanging3=(PvaluesSecond<0.05);
147 Var_Name3=CorrelatedVarNamesF2(PosSignChanging3) '
148 for i=1:size(PosSignChanging1F1)
149 if PosSignChanging1F1(i)==1 && ...
        PosSignChanging1F1(i)==PosSignChanging2(i)
150     PosTot(i)=1
151 else
152     PosTot(i)=0
153 end
154 end
155 for i=1:size(PosSignChanging1F2)
156 if PosSignChanging1F2(i)==1 && ...
        PosSignChanging1F2(i)==PosSignChanging3(i)
157     PosTot2(i)=1
158 else
159     PosTot2(i)=0
160 end
161 end
162 PosTot=logical(PosTot);
163 PosTot2=logical(PosTot2);
164 Tablestat1_reducedF1=Tablestat1_reducedF1(PosTot,:);
165 Tablestat1_reducedF2=Tablestat1_reducedF2(PosTot2,:);
166 Tablestat2_reduced=Tablestat2_reduced(PosTot,:);
167 Tablestat3_reduced=Tablestat3_reduced(PosTot2,:);
168 SignificantDatabenchmarkarrayF1= CorrelatedDatabenchmarkarrayF1(:, ...
        PosTot)
169 SignificantDatabenchmarkarrayF2=CorrelatedDatabenchmarkarrayF2(:, ...
        PosTot2);
170 SignificantDataFirstTimeLengtharray= ...
        CorrelatedDataFirstTimeLengtharray(:, PosTot);
171 SignificantDataSecondTimeLengtharray= ...
        CorrelatedDataSecondTimeLengtharray(:, PosTot2);
172 [rowsF1 columnsF1]=size(Tablestat1_reducedF1);
173 [rowsF2 columnsF2]=size(Tablestat1_reducedF2);
174 %%
175 %%
176
177 %Trend Analysis
178 [TableSTATSurrogates,K1,K2]=TrendAnalysis(Tablestat1_reducedF1,...

```

```

179     Tablestat2_reduced, Tablestat1_reducedF2, Tablestat3_reduced);
180 %%
181 SurrogateFeatureF1=SignificantDatabenchmarkarrayF1(:,K1);%Subset of ...
    Surrogate Features for first time scale investigated
182 SurrogateFeatureF2=SignificantDatabenchmarkarrayF2(:,K2);%Subset of ...
    Surrogate Features for second time scale investigated
183 VarNames2=Var_Name2(K1);
184 VarNames3=Var_Name3(K2);
185 SurrogateFeatureF1=array2table(SurrogateFeatureF1);
186 SurrogateFeatureF1.Properties.VariableNames=VarNames2;
187 SurrogateFeatureF2=array2table(SurrogateFeatureF2);
188 SurrogateFeatureF2.Properties.VariableNames=VarNames3;
189 SurrogateFeatureMatrices={SurrogateFeatureF1, SurrogateFeatureF2};
190 end

1 function [TAB1, Pvalues]=Stat(DATA,VarNames,condition, x)
2 %%This function computes the statistical analysis. As input, it ...
    takes the
3 %%matrix with observations as rows and features (predictors) as ...
    columns, the states or
4 %%labels (in numerical values) as last column. The second input, the ...
    condition, will help understand if parametric or
5 %%non-parametric analysis needs to be performed. Parametric test is
6 %%performed using t-test whereas non-parametric test is performed using
7 %%Wilcoxon rank test. The outputs are a matrix with statistical indices and
8 %%the p-value between two different conditions.
9
10 %%
11 [rows columns]=size(DATA);
12 switch x
13 case 'no' % if two conditions
14     %Find positions of the two different conditions
15     Pos_Features_Experiment=find(DATA(:,end));
16     Pos_Features_Rest=find(DATA(:,end)==0);
17
18     if ~condition % if the condition is false; positive condition is ...
        that the data are normally distributed
19         for i=1:(columns-1)
20             [p(i),h(i)]=ranksum(DATA(Pos_Features_Experiment,i),...
21                 DATA(Pos_Features_Rest,i));
22         end
23         median_Experiment=median(DATA(Pos_Features_Experiment,1:(columns-1)));
24         SD_Experiment=std(DATA(Pos_Features_Experiment,1:(columns-1)));
25         Per_Experiment=prctile(DATA...
26             (Pos_Features_Experiment,1:(columns-1)),[25 50 75]);

```

```

27
28     median_Rest=median(DATA(Pos.Features_Rest,1:(columns-1)))';
29     SD_Rest=std(DATA(Pos.Features_Rest,1:(columns-1)))';
30     Per_Rest=prctile(DATA(Pos.Features_Rest,1:(columns-1)), [25 50 ...
31         75])';
32
33     Pvalues=p(1:(columns-1))';
34
35     TAB1=table(VarNames, median_Rest, SD_Rest, Per_Rest, ...
36         median_Experiment, SD_Experiment, Per_Experiment, Pvalues);
37
38 else % if the condition is true
39     for i=1:(columns-1)
40         [h(i), p(i)]= ttest(DATA(Pos.Features_Experiment,i),...
41             DATA(Pos.Features_Rest,i));
42     end
43     %%
44     %Generate Table
45     mean_Experiment=mean(DATA(Pos.Features_Experiment,1:(columns-1)))';
46     SD_Experiment=std(DATA(Pos.Features_Experiment,1:(columns-1)))';
47     Per_Experiment=prctile(DATA...
48         (Pos.Features_Experiment,1:(columns-1)), [25 50 75])';
49
50     mean_Rest=mean(DATA(Pos.Features_Rest,1:(columns-1)))';
51     SD_Rest=std(DATA(Pos.Features_Rest,1:(columns-1)))';
52     Per_Rest=prctile(DATA(Pos.Features_Rest,1:(columns-1)), [25 50 ...
53         75])';
54
55     Pvalues=p(1:(columns-1))';
56
57     TAB1=table(VarNames, mean_Rest, SD_Rest, Per_Rest, ...
58         mean_Experiment, SD_Experiment, Per_Experiment, Pvalues);
59
60 end
61
62 case 'yes' %if one condition
63     if condition
64         mean_DATA=mean(DATA)';
65         SD_DATA=std(DATA)';
66         Per_DATA=prctile(DATA, [25 50 75])';
67         TAB1=table(VarNames, mean_DATA, SD_DATA, Per_DATA)
68     else
69         median_DATA=median(DATA)';
70         SD_DATA=std(DATA)';
71         Per_DATA=prctile(DATA, [25 50 75])';
72         TAB1=table(VarNames, median_DATA, SD_DATA, Per_DATA)

```

```

69     end
70 end
71 end

1 function [D1,SurrogateFeatureMatrices]=StatOneCond(DATAcorr1, ...
    DATAcorr2, VarNamescorr, Condition )
2 %%This function identifies subset of surrogate features via effect ...
    size analysis. The input is the
3 %%benchmark correlation matrix and the ultra-short time length ...
    correlation matrix. Also here the
4 %%condition is needed to understand if parametric or non-parametric
5 %%stat analysis needs to be performed. The outputs are the values of ...
    the stat analysis and
6 %%subset of surrogate features.
7 %%
8 [rows colomns]=size(DATAcorr1)
9 if Condition % If the features are normally distributed
10 for i=1:colomns
11     cod = cohend(DATAcorr2(:,i),DATAcorr1(:,i));
12     D(i)=abs(cod);
13     if D(i)≥0.3 %This threshold can be increased to 0.6
14         K(i)=1;
15     else
16         K(i)=0;
17     end
18 end
19 K=logical(K);
20 D1=D(K);
21 VarNameF=VarNamescorr(K);
22 SurrogateFeatureBenchmark=DATAcorr1(:,K); %Subset of Surrogate Features
23 SurrogateFeatureBenchmark=array2table(SurrogateFeatureBenchmark);
24 SurrogateFeatureBenchmark.Properties.VariableNames=VarNameF;
25 SurrogateFeatureUltraShortlength=DATAcorr2(:,K); %Subset of Surrogate ...
    Features
26 SurrogateFeatureUltraShortlength=...
27     array2table(SurrogateFeatureUltraShortlength);
28 SurrogateFeatureUltraShortlength.Properties.VariableNames=VarNameF;
29 SurrogateFeatureMatrices={SurrogateFeatureBenchmark, ...
    SurrogateFeatureUltraShortlength};
30 else % If the features are non-normally distributed
31     for i=1:colomns
32         D(i)=CliffDelta(DATAcorr2(:,i),DATAcorr1(:,i));
33         DModule(i)=abs(D(i));
34         if DModule(i)≥0.3
35             K(i)=1;

```

```

36     else
37         K(i)=0;
38     end
39 end
40 K=logical(K);
41 D1=D(K);
42 VarNameF=VarNamescorr(K);
43 SurrogateFeatureBenchmark=DATAcorr1(:,K);
44 SurrogateFeatureBenchmark=array2table(SurrogateFeatureBenchmark);
45 SurrogateFeatureBenchmark.Properties.VariableNames=VarNameF;
46 SurrogateFeatureUltraShortlength=DATAcorr2(:,K); %Subset of Surrogate ...
    Features
47 SurrogateFeatureUltraShortlength=...
    array2table(SurrogateFeatureUltraShortlength);
48 SurrogateFeatureUltraShortlength.Properties.VariableNames=VarNameF;
49 SurrogateFeatureMatrices={SurrogateFeatureBenchmark, ...
    SurrogateFeatureUltraShortlength};
51 end

1 function cod = cohend(x,y)
2 % cod = cohend(x,y);
3 % Computes Cohen's d for two independent groups using pooled standard ...
    deviation.
4 % x & y are two vectors
5 % remove NaNs & reformat
6 x=x(~isnan(x)); x=x(:); n1 = numel(x);
7 y=y(~isnan(y)); y=y(:); n2 = numel(y);
8
9 diff = mean(x) - mean(y);
10 s1 = var(x,0);
11 s2 = var(y,0);
12 psd = sqrt( ((n1-1)*s1+(n2-1)*s2) / (n1+n2-2) ); % pooled standard ...
    deviation
13 cod = diff ./ psd;

1 function d = CliffDelta(X,Y)
2 % Calculates Cliff's Delta function, a non-parametric effect magnitude
3 % test. .
4 % remove NaNs & reformat
5 X=X(~isnan(X)); X=X(:); n1 = numel(X);
6 Y=Y(~isnan(Y)); Y=Y(:); n2 = numel(Y);
7 % calculate length of vectors.
8 lx = length(X);

```

```

9 ly = length(Y);
10
11 % Comparison matrix. First dimension represents elements in X, the ...
    second elements in Y
12 % Values calculated as follows:
13 % mat(i,j) = 1 if X(i) > Y(j), zero if they are equal, and -1 if X(i) ...
    < Y(j)
14 mat = zeros(lx, ly);
15
16 % perform all the comparisons.
17 for i = 1:lx
18     for j = 1:ly
19         if X(i) > Y(j)
20             mat(i,j) = 1;
21         elseif Y(j) > X(i)
22             mat(i,j) = -1;
23         end
24     end
25 end
26
27 % calculate  $\Delta$ .
28 d = sum(mat(:)) / (lx * ly)

```

```

1 function [D]=correlation(DATA1, DATA2, condition, x)
2 %%This function generates correlation matrix. The input is the
3 %%benchmark matrix and the ultra-short time length matrices. Also ...
    here the
4 %%condition is needed to understand if parametric or non-parametric
5 %%correlation analysis needs to be performed. The output is a structure
6 %%with the rho values and p-values of the diagonal of the correlation
7 %%matrices computed for both conditions between the benchmark and the
8 %%ultra-short time length matrix.
9 %%
10 [rows columns]=size(DATA1);
11 switch x
12     case 'no'
13         Pos.Features.Experiment=find(DATA1(:,end));
14         Pos.Features.Rest=find(DATA1(:,end)==0);
15         Pos.Features.Experiment2=find(DATA2(:,end));
16         Pos.Features.Rest2=find(DATA2(:,end)==0);
17         %%
18         %Non-parametric correlation
19         if ~condition
20             %Experimental condition
21             [c-E ...

```

```

22         p_E]=corr(DATA1(Pos_Features_Experiment,1:columns-1),...
23         DATA2(Pos_Features_Experiment2,1:columns-1), 'Type', 'Spearman');
24         Buffer_E=(abs(c_E)>0.7).*(p_E<0.05); % Condition to be ...
           highly correlated and significant. The threshold of 0.7
           %can be changed to a ...
           more restricted one
25         M_E=Buffer_E;
26         D_E=diag(M_E);
27         rho_E=diag(c_E);
28         p_val_E=diag(p_E);
29     %Control condition
30     [c_R p_R]=corr(DATA1(Pos_Features_Rest,1:columns-1),...
31     DATA2(Pos_Features_Rest2,1:columns-1), 'Type', 'Spearman');
32     Buffer_R=(c_R>0.7).*(p_R<0.05); % Condition to be highly ...
           correlated and significant. The threshold of 0.7
           %can be changed to a ...
           more restricted
33     %one. However it must be ...
           changed in both ...
           conditions.
34
35     M_R=Buffer_R;
36     D_R=diag(M_R);
37     rho_R=diag(c_R);
38     p_val_R=diag(p_R);
39     D=struct('diagExp',D_E, 'diagControl', D_R, ...
           'rhoValuesExp',rho_E, 'pValuesExp',p_val_E, ...
           'rhoValuesControl',rho_R, 'pValuesControl', p_val_R);
40 else %Parametric correlation
41     %Experimental condition
42     [c_E p_E]=corr(DATA1(Pos_Features_Experiment,1:columns-1),...
43     DATA2(Pos_Features_Experiment2,1:columns-1), 'Type', 'Pearson');
44     Buffer_E=(c_E>0.7).*(p_E<0.05);
45     M_E=Buffer_E;
46     D_E=diag(M_E);
47     rho_E=diag(c_E);
48     p_val_E=diag(p_E);
49     %Control condition
50     [c_R p_R]=corr(DATA1(Pos_Features_Rest,1:columns-1),...
51     DATA2(Pos_Features_Rest2,1:columns-1), 'Type', 'Pearson');
52     Buffer_R=(c_R>0.7).*(p_R<0.05);
53     M_R=Buffer_R;
54     D_R=diag(M_R);
55     rho_R=diag(c_R);
56     p_val_R=diag(p_R);
57     D=struct('diagExp',D_E, 'diagControl', D_R, ...
           'rhoValuesExp',rho_E, 'pValuesExp',p_val_E, ...

```

```

        'rhoValuesControl',rho_R,'pValuesControl', p_val_R);
58     end
59     case 'yes'
60         %Parametric correlation
61         if condition
62             [c p]=corr(DATA1(:,1:columns), DATA2(:,1:columns), ...
                'Type', 'Pearson') ;
63             Buffer=(abs(c)>0.7).*(p<0.05);% rho must be above 0.7 ...
                and p must be less than 0.05!
64             Mask=Buffer;
65             D_M=diag(Mask);
66             rho=diag(c);
67             p_val=diag(p);
68             D=struct('diagMask',D_M, 'rhovalue', rho, 'p_val', p_val);
69         else
70             [c p]=corr(DATA1(:,1:columns), DATA2(:,1:columns), ...
                'Type', 'Spearman') ;
71             Buffer=(abs(c)>0.7).*(p<0.05);% rho must be above 0.7 ...
                and p must be less than 0.05!
72             Mask=Buffer;
73             D_M=diag(Mask);
74             rho=diag(c);
75             p_val=diag(p);
76             D=struct('diagMask',D_M, 'rhovalue', rho, 'p_val', p_val);
77         end
78 end
79 end

1 function [cr, fig, statsStruct]= BlandAltmanPlts(DATA1,DATA2, ...
    condition, x)
2 %%This function gerenates Bland-Altman plots. The input is the
3 %%benchmark matrix and the ultra-short time length matrix. Also here the
4 %%condition is needed to understand if parametric or non-parametric
5 %%analysis needs to be performed. "x" represents if one or two ...
    conditions are being analysed.
6 %%
7 [Rows Columns]=size(DATA1);
8 switch x
9 case 'no' %Case with two conditions (e.g., rest and stress)
10 Pos_Features_Experiment=find(DATA1(:,end));
11 Pos_Features_Rest=find(DATA1(:,end)==0);
12 Pos_Features_Experiment2=find(DATA2(:,end));
13 Pos_Features_Rest2=find(DATA2(:,end)==0);
14     if ~condition %NON parametric features
15         for i=1:Columns-1

```

```

16     tit = 'Bland-Altman Plot at control'; % figure title
17     gnames = {'units'};% insert units
18     label = {'Ultra-short Feature','Benchmark ...
19             Feature','units'}; % Names of the features
20     corrinfo = {'n','SSE','r2','eq'}; % stats to display ...
21             of correlation scatter plot
22     BAinfo = {'RPC(%)','ks'}; % stats to display on ...
23             Bland-Altman plot
24     limits = 'auto';
25     if 1 % colors for the data sets may be set as:
26         colors = 'br'; % character codes
27     else
28         colors = [0 0 1;... % or RGB triplets
29                 1 0 0];
30     end
31     [cr_R, fig, statsStruct_R] = BlandAltman...
32         (DATA2(Pos.Features.Rest2,i),...
33         DATA1(Pos.Features.Rest,i),...
34         label,tit,...
35         gnames,'corrInfo',...
36         corrinfo,'baInfo',BAinfo,...
37         'axesLimits',limits,...
38         'colors',colors,...
39         'baStatsMode','Non-parametric');
40     end
41     for i=1:Columns-1
42         tit = 'Bland-Altman Plot during experimental ...
43             condition'; % figure title
44         gnames = {'units'};% insert units
45         label = {'Ultra-short Feature','Benchmark ...
46                 Feature','units'}; % Names of the features
47         corrinfo = {'n','SSE','r2','eq'}; % stats to display ...
48             of correlation scatter plot
49         BAinfo = {'RPC(%)','ks'}; % stats to display on ...
50             Bland-Altman plot
51         limits = 'auto';
52         if 1 % colors for the data sets may be set as:
53             colors = 'br'; % character codes
54         else
55             colors = [0 0 1;... % or RGB triplets
56                     1 0 0];
57         end
58         [cr_E, fig, statsStruct_E] = BlandAltman...
59             (DATA2(Pos.Features.Experiment2,i),...

```

```

55         DATA1(Pos.Features.Experiment2,i), ...
56         label,tit, gnames,'corrInfo',corrinfo,...
57         'baInfo',BAinfo,'axesLimits',...
58         limits,'colors',colors,...
59         'baStatsMode','Non-parametric');
60     end
61 cr=struct('cr_R',cr_R, 'cr_E', cr_E);
62 statsStruct=struct('statsStruct_R', statsStruct_R,'statsStruct_E', ...
63     statsStruct_E );
64     else %normally distributed features
65     for i=1:Columns-1
66         tit = 'Bland-Altman Plot at control'; % figure title
67         gnames = {'units'};% insert units
68         label = {'Ultra-short Feature','Benchmark ...
69             Feature','units'}; % Names of the features
70         corrinfo = {'n','SSE','r2','eq'}; % stats to display ...
71             of correlation scatter plot
72         BAinfo = {'RPC(%)','ks'}; % stats to display on ...
73             Bland-Altman plot
74         limits = 'auto';
75         if 1 % colors for the data sets may be set as:
76             colors = 'br'; % character codes
77         else
78             colors = [0 0 1;... % or RGB triplets
79                 1 0 0];
80         end
81         [cr_R, fig, statsStruct_R] = BlandAltman...
82             (DATA2(Pos.Features.Rest2,i),DATA1(Pos.Features.Rest,i),...
83             label,tit, ...
84             gnames,'corrInfo',corrinfo,'baInfo',BAinfo,'axesLimits',...
85             limits,'colors',colors,'baStatsMode','Normal');
86     end
87     for i=1:Columns-1
88         tit = 'Bland-Altman Plot during experimental ...
89             condition'; % figure title
90         gnames = {'units'};% insert units
91         label = {'Ultra-short Feature','Benchmark ...
92             Feature','units'}; % Names of the features
93         corrinfo = {'n','SSE','r2','eq'}; % stats to ...
94             display of correlation scatter plot
95         BAinfo = {'RPC(%)','ks'}; % stats to display on ...
96             Bland-Altman plot
97         limits = 'auto';
98         if 1 % colors for the data sets may be set as:
99             colors = 'br'; % character codes

```

```

92         else
93             colors = [0 0 1;... % or RGB triplets
94                     1 0 0];
95         end
96
97         [cr_E, fig, statsStruct_E] = BlandAltman...
98             (DATA2(Pos.Features.Experiment2,i),...
99             DATA1(Pos.Features.Experiment2,i),...
100            label,tit, gnames,...
101            'corrInfo',corrinfo,...
102            'baInfo',BAinfo,'axesLimits',...
103            limits,'colors',colors,...
104            'baStatsMode','Normal');
105     end
106 cr=struct('cr_R',cr_R, 'cr_E', cr_E);
107 statsStruct=struct('statsStruct_R', statsStruct_R,'statsStruct_E', ...
108                    statsStruct_E );
109 end
110 case 'yes' %Only one condition (e.g., resting)
111     if condition % normally distributed
112         for i=1:Columns
113             tit = 'Bland-Altman Plot'; % figure title
114             gnames = {'units'};% insert units
115             label = {'Ultra-short Feature','Benchmark ...
116                     Feature','units'}; % Names of the features
117             corrinfo = {'n','SSE','r2','eq'}; % stats to display of ...
118                     correlation scatter plot
119             BAinfo = {'RPC(',')','ks'}; % stats to display on ...
120                     Bland-Altman plot
121             limits = 'auto';
122             if 1 % colors for the data sets may be set as:
123                 colors = 'br'; % character codes
124             else
125                 colors = [0 0 1;... % or RGB triplets
126                         1 0 0];
127             end
128
129             [cr, fig, statsStruct] = ...
130                 BlandAltman(DATA2(:,i),DATA1(:,i), ...
131                 label,tit, ...
132                 gnames,'corrInfo',corrinfo,'baInfo',BAinfo,'axesLimits',...
133                 limits,'colors',colors,'baStatsMode','Normal');
134         end
135     else % not normally distributed
136         for i=1:Columns
137             tit = 'Bland-Altman Plot'; % figure title

```

```

132         gnames = {'units'};% insert units
133         label = {'Ultra-short Feature','Benchmark ...
                  Feature','units'}; % Names of data sets
134         %corrinfo = {'n','rho'}; % stats to display of ...
                  correlation scatter plot
135         BAinfo = {'IQR'};; % stats to display on Bland-Altman plot
136         limits = 'auto'; % how to set the axes limits
137         if 1 % colors for the data sets may be set as:
138             colors = 'br'; % character codes
139         else
140             colors = [0 0 1;... % or RGB triplets
141                     1 0 0];
142         end
143
144         [cr, fig, statsStruct] = ...
                  BlandAltman(DATA2(:,i),DATA1(:,i),...
145                             label,tit, ...
                  gnames,'corrInfo',corrinfo,'baInfo',BAinfo,'axesLimits',...
146                             limits,'colors',colors,'baStatsMode','Non-parametric');
147         end
148     end
149
150 end
151 end

```

```

1 % BlandAltman - draws a Bland-Altman and correlation graph for two
2 % datasets.
3 %
4 % BlandAltman(data1, data2) - data1 and data2 have to be of the same size
5 % and can be grouped for display purposes. 3rd dimension is encoded by
6 % colors and 2nd dimension by symbols. The 1st dimension contains ...
   measurements
7 % within the groups.
8 % BlandAltman(data1, data2,label) - Names of data sets. Formats can be
9 %   - {'Name1'}
10 %   - {'Name1, 'Name2'}
11 %   - {'Name1, 'Name2', 'Units'}
12 % BlandAltman(data1, data2,label,tit,gnames) - Specifies the names of the
13 % groups for the legend.
14 %
15 % BlandAltman(fig, ...) - specify a figure handle in which to
16 % display
17 % figure in which the Bland-Altman and correlation will be displayed
18 % BlandAltman(ah, ...) - specify an axes which will be replaced by the
19 % Bland-Altman and correlation axes.

```

```

20 % rpc = BlandAltman(...) - return the coefficient of reproducibility
21 % (1.96 times the standard deviation of the differences)
22 % [rpc fig] = BlandAltman(...) - also return the figure handles
23 % [rpc fig sstruct] = BlandAltman(...) - also return the structure of
24 % statistics for the analysis
25 %
26 % BlandAltman(..., gnames, parameter, value) - call with parameter ...
    and value
27 % pairs using the following parameters:
28 %
29 % 'corrInfo' - specifies what information to display on the correlation
30 % plot as a cell of string in order of top to bottom. The following codes
31 % are available:
32 % - 'eq' - slope and intercept equation
33 % - 'r' - Pearson r-value
34 % - 'r2' - Pearson r-value squared
35 % - 'rho' - Spearman rho value
36 % - 'SSE' - sum of squared error
37 % - 'RMSE' - root mean squared error
38 % - 'n' - number of data points used
39 % {default = {'eq';'r2';'SSE';'n'} }
40 %
41 % 'baInfo' - specifies what information to display on the ...
    Bland-Altman plot
42 % similar to corrInfo, but with the following codes:
43 % - 'RPC' - reproducibility coefficient (1.96*SD)
44 % - 'LOA' - limits of agreement (1.96*SD) - same as RPC but ...
    different labelling
45 % - 'RPC(%)' - reproducibility coefficient and % of values
46 % - 'LOA(%)' - limits of agreement and % of values
47 % - 'CV' - coefficient of variation (SD of mean values in %)
48 % - 'IQR' - interquartile range.
49 % - 'RPCnp' - RPC estimate based on IQR (non-parametric statistics) ...
    where
50 %           RPCnp = 1.45*IQR - RPC (if distribution of differences is
51 %           normal).
52 %           See: Peck, Olsen and Devore, Introduction to ...
    Statistics and
53 %           Data Analysis. Nelson Education, 2011.
54 % - 'ks' - Kolmogorov-Smirnov test that difference-data is Gaussian
55 % - 'kurtosis' - Kurtosis test that difference-data is Gaussian
56 % - 'skewness' - skewness test results
57 % {default = {'RPC(%)';'CV'} }
58 %
59 % 'limits' - specifies the axes limits:
60 % - scalar - lower limit (eg. 0)

```

```

61 % - [min max] - specifies minimum and maximum
62 % - 'tight' - minimum and maximum of data.
63 % - 'auto' - plot default. {default}
64 %
65 % 'colors' - specify the order of group colors. (eg. 'brg' for blue, ...
        red, green) or
66 % RGB columns. {default = 'rbgmkcy'}
67 %
68 % 'symbols' - specify the order of symbols. (eg. 'sod.' for squares, ...
69 % circles, diamonds, dots). Alternatively can be set to 'Num' to display
70 % the subject number. {default = 'sodp^v'};
71 %
72 % 'markerSize' - set the size of the symbols on the plot (or font ...
        size if
73 % using 'Num' mode for symbols. {default is 4}
74 %
75 % 'data1Mode' - how to treat data set 1:
76 % - 'Compare' - data sets 1 and 2 are being compared. Means of data1 and
77 % data2 are used for x-coordinates on Bland-Altman. ...
        {default}
78 % - 'Truth' - data set 1 is considered a true reference by which data 2
79 % is being evaluated. Data 1 values are used for
80 % x-coordinates on Bland-Altman.
81 %
82 % 'forceZeroIntercept' - force the y-intercept of the linear fit on the
83 % correlation analysis to zero. {default is ...
        'off'}
84 %
85 % 'showFitCI' - show fit line confidence intervals on correlation plot.
86 % {default is 'off'};
87 %
88 % 'diffValueMode' - Units for differences:
89 % - 'Absolute' - same units as the data {default}
90 % - 'relative' - differences are normalized to the reference data ...
        (mean or
91 % data 1 depending on dataOneMode option).
92 % - 'percent' - same as relative, but in percent units.
93 %
94 % 'baYLimMode' - Mode for setting y-lim on BA axes.
95 % - 'Auto' - Automatically fit to the data.
96 % - 'Square' - Preserve 1:1 aspect ratio with x-axis and 0 is centered.
97 % {default}
98 %
99 % 'baStatsMode' - Statistical analysis mode for Bland-Altman ...
        (differnces).
100 % - 'Normal' - normal (Gaussian) distributed statistics

```

```

101 % - 'Gaussian' - same as 'Normal'.
102 % - 'Non-parametric' - non-parametric statistics.
103 %   * NOTE: Gaussian distribution is tested using the Kolmogorov-Smirnov
104 %           test. If the data seems to violate the assumption of
105 %           distribution type, a warning message is generated.
106
107
108 % by Ran Klein 2010 and adapted by Rossana Castaldo 2017
109
110 function [rpc, fig, stats] = BlandAltman(varargin)
111
112 [fig, data, params] = ParseInputArguments(varargin{:});
113 [cAH, baAH] = ConfigAxes(fig);
114
115 % Correlation plot
116 stats = CalcCorrelationStats(data, params);
117 PlotCorrelation(cAH, data, params);
118 params = FormatPlotAxes(cAH, data, params);
119 DisplayCorrelationStats(cAH, params, stats, data);
120
121 % Bland-Altman plot of differences plot
122 [stats, data, params] = CalcBASTats(stats, data, params);
123 params = PlotBA(baAH, data, stats, params);
124 DisplayBASTats(baAH, params, stats)
125
126 if ~isempty(params.tit)
127     h = supitle(params.tit);
128     set(h, 'interpreter', 'tex');
129 end
130
131 %addLegend(cAH, baAH, params)
132
133 rpc = stats.rpc;
134
135 %% Helper functions
136
137 function [fig, data, params] = ParseInputArguments(varargin)
138
139 % optional 1st parameter is figure handle
140 if isscalar(varargin{1}) && isequal(size(varargin{1}), [1 1]) && ...
    ishandle(varargin{1})
141     shift = 1;
142     fig = varargin{1};
143 else
144     shift = 0;
145     fig = [];

```

```

146 end
147
148 % followed by two data sets of equal size
149 data.set1 = varargin{shift+1};
150 data.set2 = varargin{shift+2};
151 s = size(data.set1);
152 if ~isequal(s, size(data.set2));
153     error('data1 and data2 must have the same size');
154 end
155
156 if nargin>=shift+3
157     label = varargin{shift+3};
158 else
159     label = '';
160 end
161 if nargin>=shift+4
162     params.tit = varargin{shift+4};
163 else
164     params.tit = '';
165 end
166 if nargin>=shift+5
167     params.gnames = varargin{shift+5};
168 else
169     params.gnames = '';
170 end
171
172 % default values
173 params.corrInfo = {'eq'; 'r2'; 'SSE'; 'n'};
174 params.baInfo = {'RPC(%)'; 'CV'};
175 params.defaultBaInfo = true;
176 params.axesLimits = 'auto';
177 params.colors = 'brgmcky';
178 params.symbols = 'sodp^v';
179 params.markerSize = 4;
180 params.data1TreatmentMode = 'Truth';
181 params.forceZeroIntercept = 'off';
182 params.showFitCI = 'off';
183 params.baYLimMode = 'Squared';
184 params.baStatsMode = 'Normal';
185 params.diffValueMode = 'Absolute';
186
187 % parse parameter value pair options
188 i = shift+6;
189 while length(varargin)>i
190     parameter = varargin{i};
191     val = varargin{i+1};

```

```

192     switch upper(parameter)
193         case 'CORRINFO'
194             if ischar(val)
195                 params.corrInfo = {val};
196             else
197                 params.corrInfo = val;
198             end
199
200         case 'BAINFO'
201             if ischar(val)
202                 params.baInfo = {val};
203             else
204                 params.baInfo = val;
205             end
206             params.defaultBaInfo = false;
207         case 'AXESLIMITS', params.axesLimits = val;
208         case 'COLORS', params.colors = val;
209         case 'SYMBOLS', params.symbols = val;
210         case 'MARKERSIZE', params.markerSize = val;
211         case 'DATA1MODE', params.data1TreatmentMode = val; % use the ...
                'Compare' mean of data1 and data2 or 'Truth' data1
212         case 'FORCEZEROINTERCEPT', params.forceZeroIntercept = val;
213         case 'SHOWFITCI', params.showFitCI = val;
214         case 'BASTATSMODE', params.baStatsMode = val;
215         case 'DIFFVALUEMODE', params.diffValueMode = val;
216         case 'BAYLIMODE', params.baYLimMode = val;
217
218
219     end % of swich statement
220     i = i+2;
221 end
222
223 switch length(s)
224     case 1
225         s = [s 1 1];
226     case 2
227         s = [s 1];
228     case 3
229     otherwise
230         error('Data have too many dimension');
231 end
232
233 % reformat data as an array of elements and store grouping number
234 params.numElementsPerGroup = s(1); % number of elements in each group
235 params.numGroups = numel(data.set1)/params.numElementsPerGroup;
236 params.numGroupsBySymbol = s(2);

```

```

237 params.numGroupsByColor = s(3);
238
239 if ~ischar(params.colors)
240     if size(params.colors,2)≠3
241         if size(params.colors,1)==3
242             params.colors = params.colors';
243         else
244             error('Colors must be specified in either character codes ...
                or RGB');
245         end
246     end
247 elseif size(params.colors,1)==1
248     params.colors = params.colors';
249 end
250 if size(params.colors,1)<params.numGroupsByColor
251     error('More groups than colors specified. Use the colors input ...
        variable to specify colors for each group.');
```

```

252 end
253 if ~strcmpi(params.symbols,'Num') && ...
    length(params.symbols)<params.numGroupsBySymbol
254     error('More subgroups than symbols specified. Use the symbols ...
        input variable to specify symbols for each subgroup, or use ...
        the ''Num'' option.');
```

```

255
256 end
257 data.set1 = reshape(data.set1, [numel(data.set1),1]);
258 data.set2 = reshape(data.set2, [numel(data.set2),1]);
259 data.mask = isfinite(data.set1) & isnumeric(data.set1) & ...
    isfinite(data.set2) & isnumeric(data.set2);
260 data.maskedSet1 = data.set1(data.mask);
261 data.maskedSet2 = data.set2(data.mask);
262
263 params = ResolveLabels(params,label);
264
265
266
267 %% Resolve labels and units
268 function params = ResolveLabels(params,label)
269 units = '';
270 if iscell(label)
271     if length(label)==1
272         params.d1Label = [label{1} '_1'];
273         params.d2Label = [label{1} '_2'];
274         params.meanLabel = label{1};
275         params.ΔLabel = ['\Delta ' label{1}];
276     elseif length(label)==2
```

```

277     params.d1Label = label{1};
278     params.d2Label = label{2};
279     params.meanLabel = ['Mean ' label{1} ' & ' label{2}];
280     params.ΔLabel = [label{2} ' - ' label{1}];
281     else % units also provided
282         units = label{3};
283         params.d1Label = [label{1} ' (' units ')'];
284         params.d2Label = [label{2} ' (' units ')'];
285         if strcmpi(params.dataTreatmentMode, 'Compare')
286             params.meanLabel = ['Mean ' label{1} ' & ' label{2} ' (' ...
                units ')'];
287         else
288             params.meanLabel = [label{2} ' (' units ')'];
289         end
290         switch upper(params.diffValueMode)
291             case 'ABSOLUTE'
292                 diffUnits = units;
293             case 'RELATIVE'
294                 diffUnits = '';
295                 params.baYLimMode = 'Auto';
296             case 'PERCENT'
297                 diffUnits = '%';
298                 params.baYLimMode = 'Auto';
299             otherwise
300                 error(['Unsupported diffValueMode ' ...
                    params.diffValueMode])
301         end
302         params.ΔLabel = [label{2} ' - ' label{1} ' (' diffUnits ')'];
303     end
304 else
305     params.d1Label = label;
306     params.d2Label = label;
307     params.meanLabel = label;
308     params.ΔLabel = ['\Delta ' label];
309 end
310
311 if isempty(units)
312     params.unitsStr = '';
313     params.diffUnitsStr = '';
314 else
315     params.unitsStr = [' ' units];
316     params.diffUnitsStr = [' ' diffUnits];
317 end
318
319 %% Initialize the axes (correlation and Bland-Altman) for display
320 function [cAH, baAH] = ConfigAxes(fig)

```

```

321 if isempty(fig)
322     fig = figure;
323     set(fig,'units','centimeters','position',[3 3 20 10],'color','w');
324     cAH = subplot(121);
325     baAH = subplot(122);
326 elseif strcmpi(get(fig,'type'),'figure')
327     cAH = subplot(121);
328     baAH = subplot(122);
329 elseif strcmpi(get(fig,'type'),'axes')
330     ah = fig;
331     pos = get(ah,'position');
332     fig = get(ah,'parent');
333     delete(ah);
334     cAH = axes('parent',fig,'position',[pos(1) pos(2) pos(3)/2 pos(4)]);
335     baAH = axes('parent',fig,'position',[pos(1)+pos(3)/2 pos(2) ...
        pos(3)/2 pos(4)]);
336 else
337     error('What in tarnation is the handle that was passed to ...
        Bland-Altman????')
338 end
339 set(cAH,'tag','Correlation Plot');
340 set(baAH,'tag','Bland Altman Plot');
341
342
343 %% Plot the correlation graph
344 function PlotCorrelation(cAH, data, params)
345 hold(cAH,'on');
346 for groupi=1:params.numGroups
347     if strcmpi(params.symbols,'Num')
348         for i=1:params.numElementsPerGroup
349             text(data.set2((groupi-1)*params.numElementsPerGroup+i),...
350                 data.set1((groupi-1)*params.numElementsPerGroup+i),num2str(i),...
351                 'parent',cAH,...
352                 'fontSize',params.markerSize,...
353                 'color',params.colors...
354                 (floor((groupi-1)/params.numGroupsBySymbol)+1,:),...
355                 'HorizontalAlignment','Center',...
356                 'VerticalAlignment','Middle');
357         end
358     else
359         if params.numGroupsByColor==1
360             marker = params.symbols(1);
361             color = params.colors(groupi,:);
362         else
363             marker = ...
                params.symbols(rem(groupi-1,params.numGroupsBySymbol)+1);

```

```

364         color = ...
           params.colors(floor((groupi-1)/params.numGroupsBySymbol)+1,:);
365     end
366     ph=plot(cAH, data.set1((groupi-1)*params.numElementsPerGroup+...
367         (1:params.numElementsPerGroup)),...
368         data.set2((groupi-1)*params.numElementsPerGroup+...
369         (1:params.numElementsPerGroup)),...
370         marker,...
371         'color',color);
372     set(ph,'markersize',params.markerSize);
373 end
374 end
375 xlabel(cAH,params.d2Label); ylabel(cAH,params.d1Label);
376
377
378 %% Calculate the statistical results for correlation analysis.
379 function stats = CalcCorrelationStats(data, params)
380 % Linear regression
381 if strcmpi(params.forceZeroIntercept,'on')
382     [stats.polyCoefs, stats.polyFitStruct] = ...
           polyfitZero(data.maskedSet1, data.maskedSet2, 1);
383 else
384     [stats.polyCoefs, stats.polyFitStruct] = polyfit(data.maskedSet1, ...
           data.maskedSet2, 1);
385 end
386 r = corrcoef(data.maskedSet2,data.maskedSet1);
387 stats.r=r(1,2);
388 stats.r2 = stats.r^2;
389 stats.rho = corr(data.maskedSet2,data.maskedSet1,'type','Spearman');
390 stats.N = sum(data.mask);
391 stats.SSE = ...
           sum((polyval(stats.polyCoefs,data.maskedSet1)-data.maskedSet2).^2);
392 stats.RMSE = sqrt(stats.SSE/(stats.N-2));
393 stats.slope = stats.polyCoefs(1);
394 stats.intercept = stats.polyCoefs(2);
395
396
397
398 function params = FormatPlotAxes(cAH, data, params)
399 if ischar(params.axesLimits)
400     if strcmpi(params.axesLimits,'Auto')
401         % Workaround - Add invisible minimum and maximum point to fix ...
           Auto axes limits (text
402         % does not count for axis('auto')
403         if strcmpi(params.symbols,'Num')
404             mindata = min( min(data.maskedSet1), min(data.maskedSet2) );

```

```

405         maxdata = max( max(data.maskedSet1), max(data.maskedSet2) );
406         ph = plot(cAH, [mindata maxdata], [mindata maxdata], '.', ...
                'Visible','on');
407     end
408     params.axesLimits = axis(cAH);
409     params.axesLimits(1) = ...
        min(params.axesLimits(1),params.axesLimits(3));
410     params.axesLimits(2) = ...
        max(params.axesLimits(2),params.axesLimits(4));
411     if strcmpi(params.symbols,'Num')
412         delete(ph);
413     end
414 elseif strcmpi(params.axesLimits,'Tight')
415     params.axesLimits(1) = min( min(data.maskedSet1), ...
        min(data.maskedSet2) );
416     params.axesLimits(2) = max( max(data.maskedSet1), ...
        max(data.maskedSet2) );
417 else
418     error(['Unknown axes limit option (' params.axesLimits ') ...
        detected.']);
419 end
420 else
421     if length(params.axesLimits)==1
422         a = axis(cAH);
423         params.axesLimits(2) = max(a(2),a(4));
424     else
425         % Do nothing
426     end
427 end
428 params.axesLimits(3) = params.axesLimits(1);
429 params.axesLimits(4) = params.axesLimits(2);
430
431 axis(cAH,params.axesLimits); axis(cAH,'square');
432
433
434 function DisplayCorrelationStats(cAH, params, stats, data)
435
436 x = linspace(params.axesLimits(1), params.axesLimits(2), 100);
437 [y, Δ] = polyconf(stats.polyCoefs, x, stats.polyFitStruct,'simopt','on');
438 plot(cAH, x, y, '-k');
439 if strcmpi(params.showFitCI,'on')
440     plot(cAH, x, y+Δ, '-', 'Color', 0.3*[1 1 1]);
441     plot(cAH, x, y-Δ, '-', 'Color', 0.3*[1 1 1]);
442 end
443 h = plot(cAH,params.axesLimits(1:2),params.axesLimits(1:2),':'); ...
    set(h,'color',[0.6 0.6 0.6]);

```

```

444 if 0 % Add 95% CI lines
445     xfit = params.axesLimits(1):(params.axesLimits(2)-...
446         params.axesLimits(1))/100:params.axesLimits(2);
447     [yfit, Δ] = polyconf(polyCoefs,xfit,S);
448     h = [plot(cAH,xfit,yfit+Δ);...
449         plot(cAH,xfit,yfit-Δ)];
450     set(h,'color',[0.6 0.6 0.6],'linestyle','-');
451 end
452 corrtext = {};
453 for i=1:length(params.corrInfo)
454     switch upper(params.corrInfo{i})
455         case 'EQ'
456             if ~strcmpi(params.forceZeroIntercept,'off')
457                 corrtext = [corrtext; ['y=' ...
458                     mynum2str(stats.slope,3,2) 'x']];
459             elseif stats.intercept≥0
460                 corrtext = [corrtext; ['y=' ...
461                     mynum2str(stats.slope,3,2) 'x+' ...
462                     mynum2str(stats.intercept,3)]];
463             else
464                 corrtext = [corrtext; ['y=' ...
465                     mynum2str(stats.slope,3,2) 'x' ...
466                     mynum2str(stats.intercept,3)]];
467             end
468         case 'R2', corrtext = [corrtext; ['r^2=' ...
469             mynum2str(stats.r^2,4)]];
470         case 'R', corrtext = [corrtext; ['r=' mynum2str(stats.r,4)]];
471         case 'RHO', corrtext = [corrtext; ['rho=' ...
472             mynum2str(stats.rho,4,4)]];
473         case 'SSE', corrtext = [corrtext; ['SSE=' ...
474             mynum2str(stats.SSE,2) params.unitsStr]];
475         case 'RMSE', corrtext = [corrtext; ['RMSE=' ...
476             mynum2str(stats.RMSE,2) params.unitsStr]];
477         case 'N', corrtext = [corrtext; ['n=' mynum2str(stats.N,4,0)]];
478     end
479 end
480 text(params.axesLimits(1)+...
481     0.01*(params.axesLimits(2)-params.axesLimits(1)),...
482     params.axesLimits(1)+...
483     0.9*(params.axesLimits(2)-params.axesLimits(1)),corrtext,'parent',cAH);
484
485 %% Calculate statistics for BA analysis
486 function [stats, data, params] = CalcBAStats(stats, data, params)
487
488 if strcmpi(params.data1TreatmentMode,'Compare')

```

```

481     data.maskedBaRefData = mean([data.maskedSet1,data.maskedSet2],2);
482     data.baRefData = mean([data.set1,data.set2],2);
483 else
484     data.maskedBaRefData = data.maskedSet2; % previous version was ...
         calaculated as RPC/mean of data.
485     data.baRefData = data.set1;
486 end
487 switch upper(params.diffValueMode)
488     case 'ABSOLUTE'
489         data.maskedDifferences = data.maskedSet2-data.maskedSet1;
490         data.differences = data.set2-data.set1;
491     case 'RELATIVE'
492         data.maskedDifferences = (data.maskedSet2-data.maskedSet1) ./ ...
             data.maskedBaRefData;
493         data.differences = (data.set2-data.set1) ./ data.baRefData;
494     case 'PERCENT'
495         data.maskedDifferences = (data.maskedSet2-data.maskedSet1) ./ ...
             data.maskedBaRefData*100;
496         data.differences = (data.set2-data.set1) ./ data.baRefData*100;
497 end
498 stats.differenceSTD = std(data.maskedDifferences);
499 stats.differenceMean = mean(data.maskedDifferences);
500 stats.differenceMedian = median(data.maskedDifferences);
501 [~, stats.differenceMeanP] = ttest(data.maskedDifferences,0);
502 stats.differenceMedianP = ranksum(data.set1,data.set2);
503 stats.rpc = 1.96*stats.differenceSTD;
504 stats.CV = ...
         100*stats.differenceSTD/mean((data.maskedSet1+data.maskedSet2)/2);
505 stats.rpcPercent = 1.96*std(data.maskedDifferences ./ ...
         data.maskedBaRefData)*100; % previous version was calaculated as ...
         RPC/mean of data.
506 stats.IQR = iqr(data.maskedDifferences);
507 stats.rpcNP = stats.IQR * 1.45; % estimate of RPC if distribution was ...
         Gaussian: see: R. Peck, C. Olsen, and J. Devore, Introduction to ...
         Statistics and Data Analysis. Nelson Education, 2011.
508
509 [~, stats.ksp] = ...
         kstest((data.maskedDifferences-stats.differenceMean)/stats.differenceSTD);
510 stats.kurtosis = kurtosis(data.maskedDifferences);
511 setat.skewness = skewness(data.maskedDifferences,1);
512
513
514 %% Plot the BA plot
515 function params = PlotBA(baAH, data, stats, params)
516 set(baAH,'units','normalized');
517 hold(baAH,'on');

```

```

518 for groupi=1:params.numGroups
519     ref = data.baRefData((groupi-1)*params.numElementsPerGroup+...
520         (1:params.numElementsPerGroup));
521     dif = data.differences((groupi-1)*params.numElementsPerGroup+...
522         (1:params.numElementsPerGroup));
523     if strcmpi(params.symbols,'Num')
524         for i=1:params.numElementsPerGroup
525             text(ref(i), dif(i), num2str(i), 'parent',...
526                 baAH, 'fontsize',params.markerSize,'color',...
527                 params.colors(floor((groupi-1)/params.numGroupsBySymbol)+1,:));
528         end
529     else
530         if params.numGroupsByColor==1
531             marker = params.symbols(1);
532             color = params.colors(groupi,:);
533         else
534             marker = ...
535                 params.symbols(rem(groupi-1,params.numGroupsBySymbol)+1);
536             color = ...
537                 params.colors(floor((groupi-1)/params.numGroupsBySymbol)+1,:);
538         end
539         ph = plot(baAH,ref,dif,marker,'color',color);
540         set(ph,'markersize',params.markerSize);
541     end
542 end
543 axis(baAH,'square')
544 xlabel(baAH,params.meanLabel); ylabel(baAH,params.dLabel);
545 % fix limits to +/- data limit
546 if strcmpi(params.baYLimMode,'Squared')
547     a = [params.axesLimits(1:2) ...
548         [-1 1]*abs(params.axesLimits(2)-params.axesLimits(1))/2];
549 else
550     a = axis(baAH);
551 end
552
553 fontsize = 8;
554 switch upper(params.baStatsMode)
555     case {'NORMAL','GAUSSIAN'}
556         plot(a(1:2),stats.differenceMean+[0 0],'k')
557         plot(a(1:2),stats.differenceMean+stats.rpc*[1 1],':k')
558         plot(a(1:2),stats.differenceMean-stats.rpc*[1 1],':k')
559         text(a(2),stats.differenceMean+stats.rpc, ...
560             [mynum2str(stats.differenceMean+stats.rpc,2)...
561              ' (+1.96SD)'], 'HorizontalAlignment','left',...

```

```

561         'VerticalAlignment','middle','fontsize',fontsize);
562     text(a(2),stats.differenceMean,[mynum2str(stats.differenceMean,2)...
563         ' [p=' mynum2str(stats.differenceMeanP,2) ...
564         ']' ],'HorizontalAlignment',...
565         'left','VerticalAlignment','middle','fontsize',fontsize);
566     text(a(2),stats.differenceMean-stats.rpc, ...
567         [mynum2str(stats.differenceMean-stats.rpc,2) ' ...
568         (-1.96SD)'],...
569         'HorizontalAlignment','left',...
570         'VerticalAlignment','middle',...
571         'fontsize',fontsize);
572 %     if ~isGaussian(stats)
573 %         warning('Bland-Altman analysis is being performed using a ...
Normal distribution assumptions, but the data does not appear to ...
be normally distributed. Consider using a non-parametric analysis ...
instead. See ''baStatsMode'' option for more details.')
574 %     end
575 case 'NON-PARAMETRIC'
576     m=a(1:2);
577     plot(a(1:2),stats.differenceMedian+[0 0], 'k')
578 %     x=stats.differenceMedian+stats.rpc*[1 1];
579 %     y=stats.differenceMedian-stats.rpc*[1 1];
580     plot(a(1:2),stats.differenceMedian+stats.rpcNP*[1 1], ':k')
581     plot(a(1:2),stats.differenceMedian-stats.rpcNP*[1 1], ':k')
582 %     h=fill([m fliplr(m)], [x fliplr(y)] , 'r')
583 %     set(h,'facealpha',.5);
584 %     h.FaceColor = [0.5 0.5 0.5];
585     text(a(2),stats.differenceMedian+stats.rpcNP, ...
586         [mynum2str(stats.differenceMedian+stats.rpcNP,2) ...
587         ' (+1.45IQR)'], 'HorizontalAlignment','left',...
588         'VerticalAlignment',...
589         'middle','fontsize',fontsize);
590     text(a(2),stats.differenceMedian,[mynum2str(stats.differenceMedian,2)...
591         ' [p=' mynum2str(stats.differenceMedianP,2) ']' ],...
592         'HorizontalAlignment','left',...
593         'VerticalAlignment',...
594         'middle','fontsize',fontsize);
595     text(a(2),stats.differenceMedian-stats.rpcNP,...
596         [mynum2str(stats.differenceMedian-stats.rpcNP,2) ' ...
597         (+1.45IQR) ' ],...
598         'HorizontalAlignment','left',...
599         'VerticalAlignment','middle','fontsize',fontsize);
600 %     if isGaussian(stats)
601 %         warning('Bland-Altman analysis is being performed using a ...
non-parametric distribution assumptions, but the data appears to ...
be normally distributed. Consider using a Gaussian analysis ...

```

```

instead. See 'baStatsMode' option for more details.')
599 %         end
600         % Change BA summary data overlay to RPCnp if specific overlay was
601         % not specified in the input arguments.
602         if params.defaultBaInfo
603             params.baInfo = {'RPCnp'};
604         end
605     otherwise
606         error(['Unrecognized naStatsMode ' params.baStatsMode])
607 end
608
609
610 %% Display the stats of interest on the BA plot
611 function DisplayBAStats(baAH, params, stats)
612 BAtext = {};
613 for i=1:length(params.baInfo)
614     switch upper(params.baInfo{i})
615         case 'RPC', BAtext = [BAtext; ['{\bfRPC: ' ...
            mynum2str(stats.rpc,2) params.diffUnitsStr '}']];
616         case 'RPC(%)', BAtext = [BAtext; ['{\bfRPC: ' ...
            mynum2str(stats.rpc,2) params.diffUnitsStr '}' (' ...
            mynum2str(stats.rpcPercent,2) '%')]];
617         case 'LOA', BAtext = [BAtext; ['{\bfLOA: ' ...
            mynum2str(stats.rpc,2) params.diffUnitsStr '}']];
618         case 'LOA(%)', BAtext = [BAtext; ['{\bfLOA: ' ...
            mynum2str(stats.rpc,2) params.diffUnitsStr '}' (' ...
            mynum2str(stats.rpcPercent,2) '%')]];
619         case 'CV', BAtext = [BAtext; ['CV: ' mynum2str(stats.CV,2) '%']];
620         case 'RPCNP', BAtext = [BAtext; ['{\bfRPC-{\np}: ' ...
            mynum2str(stats.rpcNP,2) params.diffUnitsStr '}']];
621         case 'KS' % Kolmogorov-Smirnov test that difference-data is ...
            Gaussian
622             BAtext = [BAtext; ['KS p-value: ' mynum2str(stats.ksp,3,3)]];
623         case 'KURTOSIS' % Kurtosis test that difference-data is Gaussian
624             BAtext = [BAtext; ['kurtosis: ' ...
            mynum2str(stats.kurtosis,2,2)]];
625         case 'SKEWNESS'
626             BAtext = [BAtext; ['skewness: ' ...
            mynum2str(stats.skewness,2,2)]];
627     end
628 end
629 a = axis(baAH);
630 text(a(2),a(4),BAtext,'interpreter',...
631     'tex','HorizontalAlignment','right',...
632     'VerticalAlignment','top','Parent',baAH);
633

```

```

634
635 %% Add legend to the plot
636 function addLegend(cAH, baAH, params)
637 gnames = params.gnames;
638 if ~strcmpi(params.symbols, 'Num') && ~isempty(gnames)
639     lh = legend('show');
640     if iscell(gnames)
641         if length(gnames)==2
642             if iscell(gnames{1})
643                 temp = cell(1,params.numGroups);
644                 for groupi=1:length(gnames{1})
645                     for j=1:length(gnames{2})
646                         temp{groupi+(j-1)*length(gnames{1})} = ...
                            [gnames{1}{groupi} '-' gnames{2}{j}];
647                     end
648                 end
649                 gnames = temp;
650             elseif iscell(gnames{2})
651                 gnames = strcat(gnames{1}, '-', gnames{2});
652             end
653         end
654     end
655     cpos = get(cAH, 'Position');
656     dpos = get(baAH, 'Position');
657     set(cAH, 'Position', cpos+[0 0.07 0 0]);
658     set(baAH, 'Position', dpos+[0 0.07 0 0]);
659     set(lh, 'string', gnames, 'orientation', 'horizontal');
660     drawnow;
661     set(lh, 'units', 'normalized');
662     pos = get(lh, 'position'); pos = min(pos(3), 0.9);
663     set(lh, 'position', [(1-pos)/2 0.02 pos 0.05]);
664 end
665
666 %% Is the BA stats data Gaussian?
667 function answer = isGaussian(stats)
668 answer = stats.ksp > 0.05;

1 % str = mynum2str(data, sigfig, maxdec) - converts a number to a string
2 %
3 % mynum2str examples (sigfig=3 and maxdec=2)
4 % 123456.0 --> 123000
5 % 123.4560 --> 123
6 % 12.34560 --> 12.3
7 % 1.234560 --> 1.23
8 % 0.1234560 --> 0.12

```

```

9
10
11 function str = mynum2str(data, sigfig, maxdec)
12 if isempty(data) || ~isfinite(data)
13     str = 'NA';
14     return
15 end
16
17 if nargin<2
18     sigfig = 3; % number of significant figures to display (before ...
                % and after .)
19 end
20 if nargin<3
21     maxdec = 2; % maximum of decimals (after the .)
22 end
23
24 if isempty(sigfig)
25     sigfig = floor(log10(abs(data)))+ 1 + maxdec;
26 end
27 e = max(-sigfig+1, floor(log10(abs(data))));
28 prec = max(-maxdec, e-sigfig+1); % precision
29 p = round(data/10^(prec))*10^(prec);
30 str = num2str(p);
31 i = find(str=='.');
32 % number of figures after the decimal
33 if isempty(i)
34     i = length(str)+1;
35     na = 0;
36 else
37     na = length(str)-i;
38 end
39 % number of figures before the decimal
40 t = str(1:i-1);
41 if strcmpi(t, '0')
42     nb = 0; % a preceding zero doesn't count
43 else
44     nb = sum(t>='0' & t<='9');
45 end
46
47 % number of trailing zeros to add
48 n = min(sigfig-(na+nb), ...
49     maxdec - na);
50 if n>0
51     if i>length(str)
52         str = [str '.'];
53     end

```

```

54     str = [str '0'*ones(1,n)];
55 end

1 function [p,S,mu] = polyfitZero(x,y,degree)
2 % POLYFITZERO Fit polynomial to data, forcing y-intercept to zero.
3 %   P = POLYFITZERO(X,Y,N) is similar POLYFIT(X,Y,N) except that the
4 %   y-intercept is forced to zero, i.e. P(N) = 0. In the same way as
5 %   POLYFIT, the coefficients, P(1:N-1), fit the data Y best in the ...
6 %   least-
7 %   squares sense. You can also use Y = POLYVAL(PZERO,X) to evaluate the
8 %   polynomial because the output is the same as POLYFIT.
9 %   [P,S,MU] = POLYFITZERO() Return structure, S, similar to POLYFIT ...
10 %   for use
11 %   with POLYVAL to calculate error estimates of predictions with P.
12 %   [P,S,MU] = POLYFITZERO() Scale X by std(X), returns MU = [0,std(X)].
13 %
14 %   See also POLYVAL, POLYFIT
15 %
16
17 % Copyright (c) 2013 Mark Mikofski
18 %% check args
19 % X & Y should be numbers
20 assert(isnumeric(x) && isnumeric(y),'polyfitZero:notNumeric', ...
21        'X and Y must be numeric.')
22 dim = numel(x); % number of elements in X
23 % DEGREE should be scalar positive number between 1 & 10 inclusive
24 assert(isnumeric(degree) && isscalar(degree) && degree>0 && ...
25        degree<=10, ...
26        'polyfitZero:degreeOutOfRange', ...
27        'DEGREE must be an integer between 1 and 10.')
28 % DEGREE must be less than number of elements in X & Y
29 assert(degree<dim && degree==round(degree), ...
30        'polyfitZero:DegreeGreaterThanDim', 'DEGREE must be less than ...
31        numel(X)')
32 % X & Y should be same size vectors
33 assert(isvector(x) && isvector(y) && dim==numel(y), ...
34        'polyfitZero:vectorMismatch', 'X and Y must be vectors of the ...
35        same length.')
36 %% solve
37 % convert X & Y to column vectors
38 x = x(:); y = y(:);
39 % Scale X.
40 % attribution: this is based on code from POLYFIT by The MathWorks Inc.

```

```

38 if nargin > 2
39     mu = [0; std(x)];
40     x = (x - mu(1))/mu(2);
41 end
42 % using pow() is actually as fast or faster than looping, same # of ...
    flops!
43 z = zeros(dim,degree);
44 for n = 1:degree
45     z(:,n) = x.^(degree-n+1);
46 end
47 p = z\y; % solve
48 p = [p;0]; % set y-intercept to zero
49 %% error estimates
50 % attribution: this is based on code from POLYFIT by The MathWorks Inc.
51 if nargin > 1
52     V = [z,ones(dim,1)]; % append constant term for Vandermonde matrix
53     % Return upper-triangular factor of QR-decomposition for error ...
        estimates
54     R = triu(qr(V,0));
55     r = y - V*p;
56     S.R = R(1:size(R,2),:);
57     S.df = max(0,length(y) - (degree+1));
58     S.normr = norm(r);
59 end
60 p = p'; % polynomial output is row vector by convention
61 end

1 function [TableSurrogates,K1,K2]=...
2     TrendAnalysis(Tablestat1.reducedF1,...
3     Tablestat2.reduced,...
4     Tablestat1.reducedF2,...
5     Tablestat3.reduced)
6 %%This function identifies the trends of the investigated features in ...
    case of two different contions (e.g., rest and stress).
7 %%
8 [rowsF1 columnsF1]=size(Tablestat1.reducedF1);
9 [rowsF2 columnsF2]=size(Tablestat1.reducedF2);
10 a=string('+');
11 b=string('-');
12 for i=1:rowsF1
13     if Tablestat1.reducedF1{i,2}-Tablestat1.reducedF1{i,5}<0
14         trendF1(i)=1;
15         trendColF1(i)=a;
16     else
17         if Tablestat1.reducedF1{i,2}-Tablestat1.reducedF1{i,5}>0

```

```

18         trendF1(i)=2;
19         trendColF1(i)=b;
20     else
21         trendF1(i)=3;
22     end
23 end
24 if Tablestat2_reduced{i,2}-Tablestat2_reduced{i,5}<0
25     trend2(i)=1;
26
27 else
28     if Tablestat2_reduced{i,2}-Tablestat2_reduced{i,5}>0
29         trend2(i)=2;
30
31     else
32         trend2(i)=3;
33     end
34 end
35 if trendF1(i)==trend2(i)
36     PosTotF1F(i)=1;
37 else
38     PosTotF1F(i)=0;
39 end
40 end
41 for i=1:rowsF2
42     if Tablestat1_reducedF2{i,2}-Tablestat1_reducedF2{i,5}<0
43         trendF2(i)=1;
44         trendColF2(i)=a;
45     else
46         if Tablestat1_reducedF2{i,2}-Tablestat1_reducedF2{i,5}>0
47             trendF2(i)=2;
48             trendColF2(i)=b;
49         else
50             trendF2(i)=3;
51         end
52     end
53     if Tablestat3_reduced{i,2}-Tablestat3_reduced{i,5}<0
54         trend3(i)=1;
55
56     else
57         if Tablestat3_reduced{i,2}-Tablestat3_reduced{i,5}>0
58             trend3(i)=2;
59
60         else
61             trend3(i)=3;
62         end
63     end

```

```

64     if trendF2(i)==trend3(i)
65         PosTotF2F(i)=1;
66     else
67         PosTotF2F(i)=0;
68     end
69 end
70 K1=logical(PosTotF1F);
71 K2=logical(PosTotF2F);
72 trendColF1=trendColF1(K1)';
73 trendColF1=table(trendColF1);
74 trendColF2=trendColF2(K2)';
75 trendColF2=table(trendColF2);
76 Tablestat2_reduced=[Tablestat2_reduced(K1,:),trendColF1];
77 Tablestat3_reduced=[Tablestat3_reduced(K2,:),trendColF2];
78 TableSurrogates={Tablestat2_reduced,Tablestat3_reduced};
79
80 end

```

A.3 Matlab tool to develop an automatic classifier using small balanced datasets

The Matlab tool to develop an automatic classifier using small balanced datasets is reported in this section.

```

1 function[FinalModel, CPFinalModel, AUCFinal, ...
    BestFeaturesPosFinalModel]= main()
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function calls the main function to develop a binary ...
    classifier for
4 %%small datasets.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 %%The output of this tool is: Final Model, Confusio Matrix, AUC, and the
7 %%features used to develop the classifier.
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9 %%Created by Rossana Castaldo, Univeristy of Warwick, 2016
10 %%Revised 2017
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 clc
13 clear all
14 close all
15 %% Splitting Dataset into Folder 1 and 2
16 [DATAFolder1, DATAFolder2]=SplittingDataset();
17 %%

```

```

18 %% Feature Selection
19 display('Feature Selection...')
20 [BestComb, BestCombPos]=FeatureSelectionProcess(DATAFolder1);
21 save('BestComb.mat');
22
23 %% Generate matrices for training
24 display('generate matrixes for training...')
25 trainingData = generateTablestraining(BestCombPos, DATAFolder1);
26
27 %% Training and validating classifiers
28 display('running classifier Tree...')
29 [trainedClassifierTree, validationAccuracyTree, CPTree, AUCTree] = ...
    trainClassifierTree(trainingData)
30 %display('saving classifier and performances for Random Forest...')
31 %save('Class_Perf_Tree','trainedClassifierTree', ...
    'validationAccuracyTree','CPTree' );
32 [trainedClassifierFinalTree, CPTree, ValAUCFinalTree, ...
    BestFeaturesPosTree]=ClassifierSelection(BestCombPos, ...
    AUCTree,CPTree, trainedClassifierTree);
33
34 display('running classifier LDA...')
35 [trainedClassifierLDA, validationAccuracyLDA,CPLDA, AUCLDA] = ...
    trainClassifierLDA(trainingData);
36 %display('saving classifier and performances for LDA...')
37 %save('Class_Perf_LDA','trainedClassifierLDA', ...
    'validationAccuracyLDA','CPRF');
38 [trainedClassifierFinalLDA, CPLDA, ValAUCFinalLDA, ...
    BestFeaturesPosLDA]=ClassifierSelection(BestCombPos, ...
    AUCLDA,CPLDA, trainedClassifierLDA);
39 %
40 %
41 % display('running classifier KNN...')
42 [trainedClassifierKNN, validationAccuracyKNN, CPKNN, AUCKNN] = ...
    trainClassifierKNN(trainingData);
43 % display('saving classifier and performances for KNN...')
44 % save('Class_Perf_KNN','trainedClassifierKNN', ...
    'validationAccuracyKNN', 'CPKNN' );
45 [trainedClassifierFinalKNN, CPKNN, ValAUCFinalKNN, ...
    BestFeaturesPosKNN]=ClassifierSelection(BestCombPos, AUCKNN, ...
    CPKNN, trainedClassifierKNN);
46
47 %% Testing best classifiers
48 TestingdataTree = generateTablestesting(DATAFolder2,BestFeaturesPosTree)
49 [validationTestingTree, CPTestingTree, AUCTestingTree, ...
    figTestingTree]=testing(TestingdataTree, trainedClassifierFinalTree);
50

```

```

51 TestingdataLDA = generateTablestesting (DATAFolder2,BestFeaturesPosLDA)
52 [validationTestingLDA, CPTestingLDA, AUCTestingLDA, ...
    figTestingLDA]=testing(TestingdataLDA, trainedClassifierFinalLDA);
53 %
54 TestingdataKNN = generateTablestesting (DATAFolder2,BestFeaturesPosKNN)
55 [validationTestingKNN, CPTestingKNN, AUCTestingKNN, ...
    figTestingKNN]=testing(TestingdataKNN, trainedClassifierFinalKNN);
56 %
57 AUCTestingAll={AUCTestingTree, AUCTestingLDA, AUCTestingKNN}
58 NFeaturesALL={BestFeaturesPosTree,BestFeaturesPosLDA, BestFeaturesPosKNN}
59 trainedClassifierALL={trainedClassifierFinalTree, ...
    trainedClassifierFinalLDA, trainedClassifierFinalKNN}
60 CPFinalALL={CPTestingTree,CPTestingLDA, CPTestingKNN };
61 %% Best model selection
62 [FinalModel, CPFinalModel, AUCFinal, ...
    BestFeaturesPosFinalModel]=ModelSelection(NFeaturesALL, ...
    CPFinalALL, AUCTestingAll, trainedClassifierALL)
63
64
65 end

1 function [DATAFolder1, DATAFolder2]=SplittingDataset ()
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function split the dataset in two folders. Folder 1 for feature
4 %%selection and training and Folder 2 for testing.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 display('Select dataset for feature selection...')
7 [filename, pathname] = uigetfile('*.csv', ' Please select the input ...
    file');
8 complete_path = strcat(pathname, filename);
9 a = readtable(complete_path);
10 VarNames=a.Properties.VariableNames(1:end);
11 %convert Table to array
12 DATA=table2array(a);
13 [rows columns]=size (DATA);
14
15 Nsub=rows/2; %for binary balanced problems
16 ID=DATA(:,1);
17
18 for i=1:Nsub
19     DATAOrdered{i,:}=DATA(ID==i,:);
20 end
21 DATAOrdered=cell2mat (DATAOrdered);
22
23 NsubFolder1=ceil((Nsub*60)/100); %60% in Folder1

```

```

24 DATAFolder1=DATAOrdered(1:(NsubFolder1*2), :);
25 DATAFolder1=array2table(DATAFolder1);
26 DATAFolder1.Properties.VariableNames=VarNames;
27 DATAFolder2=DATAOrdered(NsubFolder1*2:end, :);
28 DATAFolder2=array2table(DATAFolder2);
29 DATAFolder2.Properties.VariableNames=VarNames;
30 DATAFolder1;
31 DATAFolder2;

1 function [BestComb, BestGoodCombos]=FeatureSelectionProcess(a)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function generate all the best combination of features that are
4 %%relevant and non-redudant.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 %% Uncomment if you want to use this function alone.
7 % display('Select dataset for feature selection...')
8 % [filename, pathname] = uigetfile('*.csv', 'Please select the input ...
    file');
9 % complete_path = strcat(pathname, filename);
10 % a = readtable(complete_path);
11 VarNames=a.Properties.VariableNames(1:end-1);
12 %convert Table to array
13 DATA=table2array(a);
14 [rows columns]=size (DATA);
15 for i=1:columns-1
16     [h(i),p(i)] = lillietest(DATA(:,i)); %if h=1 the feature is ...
        non-normaly distributed
17 end
18 CountNONNormaly=sum(h==1);
19 if CountNONNormaly==columns
20     display('All features are non-normaly distributed')
21 else
22     if CountNONNormaly==0
23         display('Data is normally distributed')
24     else
25         if CountNONNormaly>(columns/2)
26             display('Many features are non-normaly distributed, ...
                strongly reccomanded to use non-parametric test')
27         else
28             if CountNONNormaly<(columns/2)
29                 display('Some features are non-normaly distributed, ...
                    strongly reccomanded to use non-parametric test ...
                    or apply log-transformation')
30             end
31         end

```

```

32     end
33 end
34 Condition=(CountNONNormaly≠0);
35 if Condition
36     prompt = 'Do you want to log-transform your data? Enter yes if ...
              you do or no if you do not: ';
37     str = input(prompt, 's');
38     switch str
39         case 'yes'
40             for i=1:columns
41                 DATA(:,i)=log( DATA(:,i));
42             end
43             CountNONNormaly=0;
44             Condition=(CountNONNormaly==0);
45         case 'no'
46             CountNONNormaly≠0;
47             Condition=(CountNONNormaly==0);
48         otherwise
49             display('error, please enter yes or no')
50     end
51 end
52 %% Relevance Analysis
53 [TAB1, Pvalues]=Stat (DATA,VarNames',Condition);
54 PosSignChanging=(Pvalues<0.05);
55 VarNamesSignificant=VarNames(PosSignChanging);
56 SignificantDATA=DATA(:,PosSignChanging);
57 %% Correlation
58 D=correlation(SignificantDATA, Condition);
59 %% Find the maximum number of features
60 m=MaxNumberofFeatures (DATA);
61 %% Find the best combination of relevant and non-redudant features
62 BestGoodCombos=Redundancy (D.Mask,m);
63 [d l]=size(BestGoodCombos)
64
65 for j=1:d
66     for i=1:l
67         BestComb{i}=VarNamesSignificant (BestGoodCombos{j,i});
68     end
69 end
70
71
72 end

1 function [TAB1, Pvalues]=Stat (DATA,VarNames,condition)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

3 %%This function computes the statistical analysis for the input ...
  matrix and
4 %%the p-value between two different conditions. As input, it takes the
5 %%matrix with observations as rows and features (predictors) as ...
  columns, the states or
6 %%labels (in binary values) as last column. The second input,the ...
  condition, will help understand if parametric or
7 %%not parametric analysis needs to be performed. Parametric test is
8 %%performed using t-test whereas non-parametric test is performed using
9 %%Wilcoxon rank test.
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 %%
12     [rows columns]=size(DATA);
13
14     %Find positions of the two different conditions
15     Pos.Features.Experiment=find(DATA(:,end));
16     Pos.Features.Rest=find(DATA(:,end)==0);
17
18     if ~condition % if the condition is false; positive condition is ...
        that the data are normally distributed
19         for i=1:(columns-1)
20             [p(i),h(i)]=ranksum...
21                 (DATA(Pos.Features.Experiment,i),...
22                 DATA(Pos.Features.Rest,i));
23         end
24         median.Experiment=median...
25             (DATA(Pos.Features.Experiment,1:(columns-1)))';
26         SD.Experiment=std...
27             (DATA(Pos.Features.Experiment,1:(columns-1)))';
28         Per.Experiment=prctile...
29             (DATA(Pos.Features.Experiment,1:(columns-1)), [25 50 75])';
30
31         median.Rest=median(DATA(Pos.Features.Rest,1:(columns-1)))';
32         SD.Rest=std(DATA(Pos.Features.Rest,1:(columns-1)))';
33         Per.Rest=prctile(DATA(Pos.Features.Rest,1:(columns-1)), [25 50 ...
34             75])';
35
36         Pvalues=p(1:(columns-1))';
37
38         TAB1=table(VarNames, median.Rest, SD.Rest, Per.Rest, ...
39             median.Experiment, SD.Experiment, Per.Experiment, Pvalues);
40     else % if the condition is true
41         for i=1:(columns-1)
42             [h(i),p(i)]= ttest...

```

```

43         DATA(Pos_Features_Rest,i));
44     end
45     %%
46     %Generate Table
47     mean_Experiment=mean...
48         (DATA(Pos_Features_Experiment,1:(columns-1)))';
49     SD_Experiment=std...
50         (DATA(Pos_Features_Experiment,1:(columns-1)))';
51     Per_Experiment=prctile...
52         (DATA(Pos_Features_Experiment,1:(columns-1)),...
53         [25 50 75])';
54
55     mean_Rest=mean(DATA(Pos_Features_Rest,1:(columns-1)))';
56     SD_Rest=std(DATA(Pos_Features_Rest,1:(columns-1)))';
57     Per_Rest=prctile(DATA(Pos_Features_Rest,1:(columns-1)), [25 50 ...
58         75])';
59
60     Pvalues=p(1:(columns-1))';
61
62     TAB1=table(VarNames,mean_Rest, SD_Rest, Per_Rest, ...
63         mean_Experiment, SD_Experiment, Per_Experiment, Pvalues);
64
65 end

1 function [D]=correlation(DATA, condition)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function generates correlation matrixes. The
4 %%condition is needed to understand if parametric or non-parametric
5 %%correlation analysis needs to be performed. The output is a structure
6 %%with the rho values and p-values of the diagonal of the correlation
7 %%matrixes.
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9 %%
10 [rows columns]=size(DATA)
11
12 %Not parametric correlation
13 if ~condition
14
15     [c_E ...
16         p_E]=corr(DATA(:,1:columns-1),DATA(:,1:columns-1), 'Type', 'Spearman');
17     Buffer_E=((abs(c_E)>0.7).*(p_E<0.05)); % Condition to be highly ...
18         correlated and significant. The threshold of 0.7
19         %can be changed to a more restricted one

```

```

18     Buffer_E(logical(eye(size(Buffer_E)))) = 0;
19     D=struct('Mask',Buffer_E, 'rhoValues',c_E,'pValues',p_E);
20 else %Parametric correlation
21
22     [c_E ...
        p_E]=corr(DATA(:,1:columns-1),DATA(:,1:columns-1),'Type','Pearson');
23     Buffer_E=(abs(c_E)>0.7).*(p_E<0.05);
24     Buffer_E(logical(eye(size(Buffer_E)))) = 0;
25     D=struct('Mask',Buffer_E, 'rhoValues',c_E,'pValues',p_E);
26 end
27
28
29 end

```

```

1 function n=MaxNumberOfFeatures(DATA)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function calculates the maximum number of features that the model
4 %%can contain in order to avoid overfitting
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7 [rows columns]=size(DATA);
8 Pos_Features_Experiment=find(DATA(:,end));
9 Pos_Features_Rest=find(DATA(:,end)==0);
10 s=size(Pos_Features_Experiment);
11 if ~isequal(s,size(Pos_Features_Rest));
12     error('same number of instances during the two conditions');
13 end
14 NumbofSub=length(Pos_Features_Experiment);
15 n=NumbofSub/10; %Rule of Thumb
16 n=ceil(n);
17 if n>columns
18     n=columns;
19 else
20     n=n;
21 end

```

```

1 function BestGoodCombosPos=Redundancy(M,m)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function generates all the possible combinations of features. As
4 %%input, it takes the correlation matrix (M) and the max number of ...
5 %%that a combination can contain.
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

7 n=length(M)
8 GoodCombosPos=[];
9 k=1;
10 X=1;
11 if m>n
12     m=n;
13 else
14     m=m;
15 end
16
17 for k=1:m
18     combos = nchoosek((1:n),k);
19     Ncombos=size(combos,1);
20     for i=1:Ncombos
21
22         Sums(i)=sum(sum(M(combos(i,:),combos(i,:))));
23
24     end
25
26     s(k).GoodCombosPos=find(Sums==0);
27     s(k).combos=combos;
28     Condition=(length(s(k).GoodCombosPos)>0);
29     if Condition
30
31         BestGoodCombosPos{k}=s(k).GoodCombosPos;
32         BestGoodCombos{k}=s(k).combos(s(k).GoodCombosPos,:);
33
34     end
35     clear GoodCombosPos combos Sums
36 end
37
38 end

1 function [trainingData] = generateTablestraining(BestCombPos, ...
datasettraining)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function generates the training datasets for each features ...
combination
4 %%It takes as input the data from Folder 1 and the features' ...
combinations computed in
5 %%FeatureSelectionProcess function.
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 datasettraining=datasettraining(:, 2:end); %only HRV features and class
8 VarNames=datasettraining.Properties.VariableNames
9 D=table2array(datasettraining);

```

```

10 [l p]=size(BestCombPos);
11 for j=1:p
12     [n m]=size(BestCombPos{:,j});
13     for i=1:n
14         b=BestCombPos{:,j}
15         trainingData{j,i}=[datasettraining(:,b(i,:)), datasettraining(:,end)]
16     end
17 end
18 end

1 function [trainedClassifier, validationAccuracy, CP, AUC] = ...
    trainClassifierTree(trainingData)
2 % trainClassifier(trainingData)
3 % returns a trained classifier and its accuracy.
4 % This code recreates the classification model trained in
5 % Classification Learner app.
6 %
7 % Input:
8 %     trainingData: the training data of same data type as imported
9 %     in the app (table or matrix).
10 %
11 % Output:
12 %     trainedClassifier: a struct containing the trained classifier.
13 %     The struct contains various fields with information about the
14 %     trained classifier.
15 %
16 %     trainedClassifier.predictFcn: a function to make predictions
17 %     on new data. It takes an input of the same form as this training
18 %     code (table or matrix) and returns predictions for the response.
19 %     If you supply a matrix, include only the predictors columns (or
20 %     rows).
21 %
22 %     validationAccuracy: a double containing the accuracy in
23 %     percent. In the app, the History list displays this
24 %     overall accuracy score for each model.
25 %
26 % Use the code to train the model with new data.
27 % To retrain your classifier, call the function from the command line
28 % with your original data or new data as the input argument ...
    trainingData.
29 %
30 % For example, to retrain a classifier trained with the original ...
    data set
31 % T, enter:
32 %     [trainedClassifier, validationAccuracy] = trainClassifier(T)

```

```

33 %
34 % To make predictions with the returned 'trainedClassifier' on new ...
    data T,
35 % use
36 %     yfit = trainedClassifier.predictFcn(T)
37 %
38 % To automate training the same classifier with new data, or to ...
    learn how
39 % to programmatically train classifiers, examine the generated code.
40
41
42
43 % Extract predictors and response
44 % This code processes the data into the right shape for training the
45 % classifier.
46 [m n]=size(trainingData);
47 for j=1:m
48     [k n(j)]=size(find(~cellfun(@isempty,trainingData(j,:))));
49     for i=1:n(j)
50         inputTable = trainingData{j,i};
51
52         predictorNames = inputTable.Properties.VariableNames(1:end-1);
53         predictors = inputTable(:,(predictorNames));
54         response = inputTable{: ,end};
55         response=array2table(response);
56         isCategoricalPredictor = false(1, ...
            length(predictorNames(1:end-1)));
57
58         % Train a classifier
59         % This code specifies all the classifier options and trains ...
            the classifier.
60         VariableDescriptions = ...
            hyperparameters('fitcensemble',predictors,response,'Tree');
61         classificationTree = fitcensemble(predictors, response,...
62             'OptimizeHyperparameters','auto',...
63             'HyperparameterOptimizationOptions',...
64             struct('AcquisitionFunctionName',...
65                 'expected-improvement-plus'));
66
67
68         % Create the result struct with predict function
69         predictorExtractionFcn = @(t) t(:, predictorNames);
70         treePredictFcn = @(x) predict(classificationTree, x);
71
72         trainedClassifier(j,i).predictFcn = @(x) ...
            treePredictFcn(predictorExtractionFcn(x));

```

```

73
74     % Add additional fields to the result struct
75     trainedClassifier(j,i).RequiredVariables = ...
        inputTable.Properties.VariableNames;
76     trainedClassifier(j,i).ClassificationTree = classificationTree;
77     trainedClassifier(j,i).About = 'This struct is a trained ...
        classifier exported from Classification Learner R2016a.';
78     trainedClassifier(j,i).HowToPredict = sprintf('To make ...
        predictions on a new table, T, use: \n yfit = ...
        c.predictFcn(T) \nreplacing ''c'' with the name of the ...
        variable that is this struct, e.g. ''trainedClassifier''. ...
        \n \nThe table, T, must contain the variables returned ...
        by: \n c.RequiredVariables \nVariable formats (e.g. ...
        matrix/vector, datatype) must match the original training ...
        data. \nAdditional variables are ignored. \n \nFor more ...
        information, see <a ...
        href=""matlab:helpview(fullfile(docroot, ''stats'', ...
        ''stats.map''), ...
        ''appclassification_exportmodeltoworkspace'')">How to ...
        predict using an exported model</a>.');
79
80
81     % Perform cross-validation
82     k=MaxNumCrossVal(inputTable);
83     partitionedModel = ...
        crossval(trainedClassifier(j,i).ClassificationTree, ...
        'Kfold', k);
84
85     % Compute validation accuracy
86     validationAccuracy{j,i} = 1 - kfoldLoss(partitionedModel, ...
        'LossFun', 'ClassifError');
87
88     % Compute validation predictions and scores
89     [validationPredictions, validationScores] = ...
        kfoldPredict(partitionedModel);
90     response=table2array(response);
91     C{i}= confusionmat(response,validationPredictions);
92     CP{i} = classperf(response,validationPredictions);
93     [X,Y,T,AUC{j,i},OPTROCPT,SUBY,SUBYNAMES] = ...
        perfcurve(response,validationScores(:,2),1);
94     figure, plot(X,Y,OPTROCPT(1),OPTROCPT(2),'r*'),grid on;
95     xlabel('False positive rate')
96     ylabel('True positive rate')
97     title('ROC')
98     end
99 end

```

100 end

```
1 function [trainedClassifier, validationAccuracy, CP, AUC] = ...
    trainClassifierLDA(trainingData)
2 % trainClassifier(trainingData)
3 % returns a trained classifier and its accuracy.
4 % This code recreates the classification model trained in
5 % Classification Learner app.
6 %
7 % Input:
8 %     trainingData: the training data of same data type as imported
9 %     in the app (table or matrix).
10 %
11 % Output:
12 %     trainedClassifier: a struct containing the trained classifier.
13 %     The struct contains various fields with information about the
14 %     trained classifier.
15 %
16 %     trainedClassifier.predictFcn: a function to make predictions
17 %     on new data. It takes an input of the same form as this training
18 %     code (table or matrix) and returns predictions for the response.
19 %     If you supply a matrix, include only the predictors columns (or
20 %     rows).
21 %
22 %     validationAccuracy: a double containing the accuracy in
23 %     percent. In the app, the History list displays this
24 %     overall accuracy score for each model.
25 %
26 % Use the code to train the model with new data.
27 % To retrain your classifier, call the function from the command line
28 % with your original data or new data as the input argument ...
    trainingData.
29 %
30 % For example, to retrain a classifier trained with the original ...
    data set
31 % T, enter:
32 %     [trainedClassifier, validationAccuracy] = trainClassifier(T)
33 %
34 % To make predictions with the returned 'trainedClassifier' on new ...
    data T,
35 % use
36 %     yfit = trainedClassifier.predictFcn(T)
37
38
39
```

```

40 % Extract predictors and response
41 % This code processes the data into the right shape for training the
42 % classifier.
43 [m n]=size(trainingData);
44 for j=1:m
45     [k n(j)]=size(find(~cellfun(@isempty,trainingData{j,:})));
46     for i=1:n(j)
47         inputTable = trainingData{j,i};
48
49         predictorNames = inputTable.Properties.VariableNames(1:end-1);
50         predictors = inputTable(:,(predictorNames));
51         response = inputTable{:,end};
52         response=array2table(response);
53         isCategoricalPredictor = false(1, ...
54             length(predictorNames(1:end-1)));
55
56         % Train a classifier
57         % This code specifies all the classifier options and trains ...
58         % the classifier.
59         classificationDiscriminant = fitcdiscr(...
60             predictors, ...
61             response, ...
62             'OptimizeHyperparameters','auto', ...
63             'HyperparameterOptimizationOptions',...
64             struct('AcquisitionFunctionName','expected-improvement-plus'));
65         predictorExtractionFcn = @(t) t(:, predictorNames);
66         discriminantPredictFcn = @(x) ...
67             predict(classificationDiscriminant, x);
68
69         trainedClassifier(j,i).predictFcn = @(x) ...
70             discriminantPredictFcn(predictorExtractionFcn(x));
71
72         % Add additional fields to the result struct
73         trainedClassifier(j,i).RequiredVariables = ...
74             inputTable.Properties.VariableNames;
75         trainedClassifier(j,i).ClassificationDiscriminant = ...
76             classificationDiscriminant;
77         trainedClassifier(j,i).About = 'This struct is a trained ...
78             classifier exported from Classification Learner R2016a.';
79         trainedClassifier(j,i).HowToPredict = sprintf('To make ...
80             predictions on a new table, T, use: \n yfit = ...
81             c.predictFcn(T) \nreplacing ''c'' with the name of the ...
82             variable that is this struct, e.g. ''trainedClassifier''. ...
83             \n \nThe table, T, must contain the variables returned ...
84             by: \n c.RequiredVariables \nVariable formats (e.g. ...

```

```

matrix/vector, datatype) must match the original training ...
data. \nAdditional variables are ignored. \n \nFor more ...
information, see <a ...
href="matlab:helpview(fullfile(docroot, 'stats', ...
'stats.map'), ...
'appclassification_exportmodeltoworkspace')">How to ...
predict using an exported model</a>.');
73
74 % Perform cross-validation
75 k=MaxNumbCrossVal(inputTable);
76 partitionedModel = ...
    crossval(trainedClassifier(j,i).ClassificationDiscriminant, ...
    'KFold', k);
77
78 % Compute validation accuracy
79 validationAccuracy{j,i} = 1 - kfoldLoss(partitionedModel, ...
    'LossFun', 'ClassifError');
80
81 % Compute validation predictions and scores
82 [validationPredictions, validationScores] = ...
    kfoldPredict(partitionedModel);
83 response=table2array(response);
84 C{i}= confusionmat(response,validationPredictions);
85 CP{i} = classperf(response,validationPredictions);
86 [X,Y,T,AUC{j,i},OPTROCPT,SUBY,SUBYNAMES] = ...
    perfcurve(response,validationScores(:,2),1);
87 figure, plot(X,Y,OPTROCPT(1),OPTROCPT(2),'r*'), grid on;
88 xlabel('False positive rate')
89 ylabel('True positive rate')
90 title('ROC')
91
92 end
93 end
94 end

1 function [trainedClassifier, validationAccuracy, CP, AUC] = ...
    trainClassifierKNN(trainingData)
2 % trainClassifier(trainingData)
3 % returns a trained classifier and its accuracy.
4 % This code recreates the classification model trained in
5 % Classification Learner app.
6 %
7 % Input:
8 %     trainingData: the training data of same data type as imported
9 %     in the app (table or matrix).

```

```

10 %
11 %   Output:
12 %       trainedClassifier: a struct containing the trained classifier.
13 %       The struct contains various fields with information about the
14 %       trained classifier.
15 %
16 %       trainedClassifier.predictFcn: a function to make predictions
17 %       on new data. It takes an input of the same form as this training
18 %       code (table or matrix) and returns predictions for the response.
19 %       If you supply a matrix, include only the predictors columns (or
20 %       rows).
21 %
22 %       validationAccuracy: a double containing the accuracy in
23 %       percent. In the app, the History list displays this
24 %       overall accuracy score for each model.
25 %
26 % Use the code to train the model with new data.
27 % To retrain your classifier, call the function from the command line
28 % with your original data or new data as the input argument ...
   trainingData.
29 %
30 % For example, to retrain a classifier trained with the original ...
   data set
31 % T, enter:
32 %   [trainedClassifier, validationAccuracy] = trainClassifier(T)
33 %
34 % To make predictions with the returned 'trainedClassifier' on new ...
   data T,
35 % use
36 %   yfit = trainedClassifier.predictFcn(T)
37 %
38
39
40
41
42 % Extract predictors and response
43 % This code processes the data into the right shape for training the
44 % classifier.
45 [m n]=size(trainingData);
46 for j=1:m
47     [k n(j)]=size(find(~cellfun(@isempty,trainingData(j,:))));
48     for i=1:n(j)
49         inputTable = trainingData{j,i};
50
51         predictorNames = inputTable.Properties.VariableNames;
52         predictors = inputTable(:,(predictorNames(1:end-1)));

```

```

53     response = inputTable{:,end};
54     response=array2table(response);
55     isCategoricalPredictor = false(1, ...
56         length(predictorNames(1:end-1)));
57     % Train a classifier
58     % This code specifies all the classifier options and trains ...
59     % the classifier.
60     %K=5;
61     classificationKNN = fitcknn(...
62         predictors, ...
63         response, ...
64         'OptimizeHyperparameters','auto',...
65         'HyperparameterOptimizationOptions',...
66         struct('AcquisitionFunctionName','expected-improvement-plus'))
67
68     % Create the result struct with predict function
69     predictorExtractionFcn = @(t) t(:, predictorNames);
70     knnPredictFcn = @(x) predict(classificationKNN, x);
71     trainedClassifier(j,i).predictFcn = @(x) ...
72         knnPredictFcn(predictorExtractionFcn(x));
73
74     % Add additional fields to the result struct
75     trainedClassifier(j,i).RequiredVariables = ...
76         inputTable.Properties.VariableNames;
77     trainedClassifier(j,i).ClassificationKNN = classificationKNN;
78     trainedClassifier(j,i).About = 'This struct is a trained ...
79         classifier exported from Classification Learner R2016a.';
80     trainedClassifier(j,i).HowToPredict = sprintf('To make ...
81         predictions on a new table, T, use: \n yfit = ...
82         c.predictFcn(T) \nreplacing ''c'' with the name of the ...
83         variable that is this struct, e.g. ''trainedClassifier''. ...
84         \n \nThe table, T, must contain the variables returned ...
85         by: \n c.RequiredVariables \nVariable formats (e.g. ...
86         matrix/vector, datatype) must match the original training ...
87         data. \nAdditional variables are ignored. \n \nFor more ...
88         information, see <a ...
89         href="matlab:helpview(fullfile(docroot, ''stats'', ...
90         ''stats.map''), ...
91         ''appclassification_exportmodeltoworkspace'')">How to ...
92         predict using an exported model</a>.'');
93
94     % Perform cross-validation
95     k=MaxNumbCrossVal(inputTable);
96     partitionedModel = ...

```

```

        crossval(trainedClassifier(j,i).ClassificationKNN, ...
            'KFold', k);
82     % Compute validation accuracy
83     validationAccuracy{j,i} = 1 - kfoldLoss(partitionedModel, ...
        'LossFun', 'ClassifError');

84
85     % Compute validation predictions and scores
86     [validationPredictions, validationScores] = ...
        kfoldPredict(partitionedModel);
87     response=table2array(response);
88     C{i}= confusionmat(response,validationPredictions);
89     CP{i} = classperf(response,validationPredictions);
90     [X,Y,T,AUC{j,i},OPTROCPT,SUBY,SUBYNAMES] = ...
        perfcurve(response,validationScores(:,2),1);
91     figure, plot(X,Y,OPTROCPT(1),OPTROCPT(2),'r*'), grid on;
92     xlabel('False positive rate')
93     ylabel('True positive rate')
94     title('ROC')
95
96
97     end
98 end
99 end

1 function [k]=MaxNumbCrossVal(Folder1)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function calculates the maximum number of cross-validation ...
   folders.
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 [rows columns]=size(Folder1);
6 NSub=(rows)/2;
7 k=NSub/10;
8 k=ceil(k);
9 if k==1
10     k=NSub-1;
11 else
12     k
13 end
14 end

1 function [trainedClassifierFinal, CPFinal, ValAUCFinal, ...
   BestFeaturesPos]=ClassifierSelection(BestCombPos, AUC, CP, ...
   trainedClassifier)

```

```

2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function selects the best classifier for each machine learning
4 %%methods used.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7 [l g]= size(AUC)
8
9 for i=1:g
10     [value(i),index(i)]= max([AUC{: ,i}]);
11     row(i)=i
12     col(i)=index(i);
13     values(i)=value(i);
14 end
15
16 position=[row', col', value'];
17 [ValAUCFinal,ind]=max(position(:,3));
18 IndexFinal=position(ind,1:end-1);
19
20 CombPos=BestCombPos(IndexFinal(1,1),1);
21 CombPosM=cell2mat(CombPos);
22 BestFeaturesPos=CombPosM(IndexFinal(1,2),:);
23 CPFinal=CP(IndexFinal(1,2));
24
25 trainedClassifierFinal=trainedClassifier(IndexFinal(1,1),IndexFinal(1,2))
26 end

1 function [validation, C, AUC, fig]=testing(Testingdata, ...
    trainedClassifier)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function tests the best classifier on Folder 2 and gives the ...
    accuracy
4 %%performance, confusio matrix, AUC value and the ROC curve.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7 yfit = trainedClassifier.predictFcn(Testingdata);
8 response=Testingdata(:,end);
9 response=table2array(response);
10 [row column]=size(yfit);
11 correctprediction=zeros(row,1);
12 for i=1:row
13     if yfit (i)==response(i)
14         correctprediction(i)=1;
15     else
16         correctprediction(i)=0;
17     end

```

```

18 end
19 validation=sum(correctprediction)/length(correctprediction);
20 C = confusionmat(response,yfit);
21 CP = classperf(response,yfit);
22
23
24 [X,Y,T,AUC,OPTROCPT,SUBY,SUBYNAMES] = perfcurve(response,yfit,1);
25 fig= plot(X,Y,OPTROCPT(1),OPTROCPT(2),'r*'), grid on;
26 xlabel('False positive rate')
27 ylabel('True positive rate')
28 title('ROC')
29
30 end

1 function [trainedClassifierFinal, CPFinalModel, AUCT, ...
    BestFeaturesPos]=ModelSelection(NFeaturesALL, CP, AUCTestingAll, ...
    trainedClassifierALL)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function selects among the different machine learning methods the
4 %%best classifier: max AUC and the lowest number of features used.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 [l g]= size(AUCTestingAll)
7
8 for i=1:l
9     [value(i),index(i)]= max([AUCTestingAll{: ,i}]);
10    row(i)=i;
11    col(i)=index(i);
12    values(i)=value(i);
13    %position=[row, col, value];
14 end
15
16 [m n]=size(NFeaturesALL);
17 for j=1:m
18     L(j)=length([NFeaturesALL{: ,j}]);
19     [valueL(j),indexL(j)]=min(L(j));
20     rowL(j)=j;
21     colL(j)=indexL(j);
22     valuesL(j)=valueL(j);
23     %position=[row, col, value];
24 end
25 position=[row', col', value'];
26 [AUCTFinal,ind]=max(position(:,3));
27 IndexFinal=position(ind,1:end-1);
28
29 positionL=[rowL', colL', valueL'];

```

```

30 [L, indL]=min(positionL(:,3));
31 IndexFinalL=positionL(indL,1:end-1);
32 AUCFinalL=AUCTestingAll(:, IndexFinalL(1,2));
33
34
35 CombPos=NFeaturesALL(IndexFinal(1,1),1);
36 CombPosM=cell2mat(CombPos);
37 BestFeaturesPos=CombPosM(IndexFinal(1,2),:);
38 AUCFinalL=cell2mat(AUCFinalL);
39 %AUCFinal=cell2mat(AUCFinal)
40 if AUCFinalL<=AUCFinal
41     AUCT=AUCFinal;
42     Index=IndexFinal;
43 else
44     AUCT=AUCFinalL
45     Index=IndexFinalL;
46 end
47 trainedClassifierFinal=trainedClassifierALL(Index(1,1), Index(1,2))
48 CPFinalModel=CP(Index(1,1), Index(1,2));
49 end

```

A.4 Matlab tool to develop an automatic classifier using unbalanced datasets

The Matlab tool to develop an automatic classifier using unbalanced datasets is reported in this section.

```

1 function[FinalModel, CPFinalModel, AUCFinal, ...
    BestFeaturesPosFinalModel]= main()
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function calls the main function to develop a binary ...
    classifier for
4 %%unbalanced datasets.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 %%The output of this tool is: Final Model, Confusio Matrix, AUC, and the
7 %%features used to develop the classifier.
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9 %%Created by Rossana Castaldo, Univeristy of Warwick, 2016
10 %%Revised 2017
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 clc
13 clear all
14 close all

```

```

15 %% Splitting Dataset into Folder 1, 2 and 3
16 [DATAFolder1, DATAFolder2, DATAFolder3]=SplittingDataset();
17 %%
18 %% Feature Selection
19 display('Feature Selection...')
20 [BestComb, BestCombPos]=FeatureSelectionProcess(DATAFolder1);
21 save('BestComb.mat');
22
23 %% Generate matrices for training
24 display('generate matrixes for training...')
25 trainingData = generateTablestraining(BestCombPos, DATAFolder2);
26
27 %% Training and validating classifiers
28 display('running classifier Tree...')
29 N=MinNumbN(DATAFolder2);
30 [trainedClassifierTree, validationAccuracyTree, CPTree, AUCTree] = ...
    trainClassifierTree(trainingData,N);
31 %[trainedClassifierTree, validationAccuracyTree, CPTree, AUCTree] = ...
    trainClassifierTree(trainingData,N);
32 %display('saving classifier and performances for Random Forest...')
33 %save('Class_Perf_Tree','trainedClassifierTree', ...
    'validationAccuracyTree','CPTree' );
34 [trainedClassifierFinalTree, CPTree, ValAUCFinalTree, ...
    BestFeaturesPosTree]=ClassifierSelection(BestCombPos, ...
    AUCTree,CPTree, trainedClassifierTree);
35
36 display('running classifier LDA...')
37 [trainedClassifierLDA, validationAccuracyLDA,CPLDA, AUCLDA] = ...
    trainClassifierLDA(trainingData, N);
38 %display('saving classifier and performances for LDA...')
39 %save('Class_Perf_LDA','trainedClassifierLDA', ...
    'validationAccuracyLDA','CPRF');
40 [trainedClassifierFinalLDA, CPLDA, ValAUCFinalLDA, ...
    BestFeaturesPosLDA]=ClassifierSelection(BestCombPos, ...
    AUCLDA,CPLDA, trainedClassifierLDA);
41 % %
42 % %
43 display('running classifier KNN...')
44 [trainedClassifierKNN, validationAccuracyKNN, CPKNN, AUCKNN] = ...
    trainClassifierKNN(trainingData, N);
45 % display('saving classifier and performances for KNN...')
46 % save('Class_Perf_KNN','trainedClassifierKNN', ...
    'validationAccuracyKNN', 'CPKNN' );
47 [trainedClassifierFinalKNN, CPKNN, ValAUCFinalKNN, ...
    BestFeaturesPosKNN]=ClassifierSelection(BestCombPos, AUCKNN, ...
    CPKNN, trainedClassifierKNN);

```

```

48
49 %% Testing best classifiers
50 TestingdataTree = generateTablestesting(DATAFolder3,BestFeaturesPosTree)
51 [validationTestingTree, CPTestingTree, AUCTestingTree, ...
    figTestingTree]=testing(TestingdataTree, trainedClassifierFinalTree);
52 %
53 TestingdataLDA = generateTablestesting(DATAFolder3,BestFeaturesPosLDA)
54 [validationTestingLDA, CPTestingLDA, AUCTestingLDA, ...
    figTestingLDA]=testing(TestingdataLDA, trainedClassifierFinalLDA);
55 %
56 TestingdataKNN = generateTablestesting(DATAFolder3,BestFeaturesPosKNN)
57 [validationTestingKNN, CPTestingKNN, AUCTestingKNN, ...
    figTestingKNN]=testing(TestingdataKNN, trainedClassifierFinalKNN);
58 %
59 AUCTestingAll={AUCTestingTree, AUCTestingLDA, AUCTestingKNN}
60 NFeaturesALL={BestFeaturesPosTree,BestFeaturesPosLDA, BestFeaturesPosKNN}
61 trainedClassifierALL={trainedClassifierFinalTree, ...
    trainedClassifierFinalLDA, trainedClassifierFinalKNN}
62 CPFinalALL={CPTestingTree,CPTestingLDA, CPTestingKNN };
63 %% Best model selection
64 [FinalModel, CPFinalModel, AUCFinal, ...
    BestFeaturesPosFinalModel]=ModelSelection(NFeaturesALL, ...
    CPFinalALL, AUCTestingAll, trainedClassifierALL)
65
66
67 end

1 function [DATAFolder1, DATAFolder2, DATAFolder3]=SplittingDataset()
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function splits the dataset into 3 folders. Folder 1 for feature
4 %%selection, Folder 2 for training and validation, and Folder 3 for ...
    testing.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 display('Select dataset for feature selection...')
7 [filename, pathname] = uigetfile('*.*', ' Please select the input ...
    file');
8 complete_path = strcat(pathname, filename);
9 a = readtable(complete_path);
10 VarNames=a.Properties.VariableNames(1:end);
11 %convert Table to array
12 DATA=table2array(a);
13 [rows columns]=size (DATA);
14
15 Pos_Features_Experiment=find(DATA(:,end)==1);
16 lE=length(Pos_Features_Experiment);

```

```

17 Pos_Features_Control=find(DATA(:,end)==0);
18 lC=length(Pos_Features_Control);
19 NsubEFolder1=ceil((lE*34)/100);
20 NsubCFolder1=ceil((lC*34)/100);
21 NsubEFolder2=ceil((lE*39)/100);
22 NsubCFolder2=ceil((lC*39)/100);
23 NsubEFolder3=ceil((lE*27)/100);
24 NsubCFolder3=ceil((lC*27)/100);
25
26 DATAFolder1E=DATA(Pos_Features_Experiment(1:NsubEFolder1), :);
27 DATAFolder1C=DATA(Pos_Features_Control(1:NsubCFolder1), :);
28 DATAFolder1=[DATAFolder1E; DATAFolder1C];
29 DATAFolder1=array2table(DATAFolder1);
30 DATAFolder1.Properties.VariableNames=VarNames;
31
32 DATAFolder2E=...
33     DATA(Pos_Features_Experiment...
34         (NsubEFolder1:(NsubEFolder1+NsubEFolder2)), :);
35 DATAFolder2C=...
36     DATA(Pos_Features_Control...
37         (NsubCFolder1:(NsubCFolder1+NsubCFolder2)), :);
38 DATAFolder2=[DATAFolder2E; DATAFolder2C];
39 DATAFolder2=array2table(DATAFolder2);
40 DATAFolder2.Properties.VariableNames=VarNames;
41
42
43 DATAFolder3E=...
44     DATA(Pos_Features_Experiment...
45         (NsubEFolder2:(NsubEFolder2+NsubEFolder3)), :);
46 DATAFolder3C=...
47     DATA(Pos_Features_Control...
48         (NsubCFolder2:(NsubCFolder2+NsubCFolder3)), :);
49 DATAFolder3=[DATAFolder3E; DATAFolder3C];
50 DATAFolder3=array2table(DATAFolder3);
51 DATAFolder3.Properties.VariableNames=VarNames;
52
53
54
55 DATAFolder1;
56 DATAFolder2;
57 DATAFolder3;

1 function [BestComb, BestGoodCombos]=FeatureSelectionProcess(a)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function generates all the best combination of features that are

```

```

4 %%relevant and non-redudant.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 %% Uncomment if you want to use this function alone.
7 % display('Select dataset for feature selection...')
8 % [filename, pathname] = uigetfile('*.csv', ' Please select the input ...
    file');
9 % complete_path = strcat(pathname, filename);
10 % a = readtable(complete_path);
11 VarNames=a.Properties.VariableNames(1:end-1);
12 %convert Table to array
13 DATA=table2array(a);
14 [rows columns]=size (DATA);
15 for i=1:columns-1
16     [h(i),p(i)] = lillietest(DATA(:,i)); %if h=1 the feature is ...
        non-normaly distributed
17 end
18 CountNONNormaly=sum(h==1);
19 if CountNONNormaly==columns
20     display('All features are non-normaly distributed')
21 else
22     if CountNONNormaly==0
23         display('Data is normally distributed')
24     else
25         if CountNONNormaly>(columns/2)
26             display('Many features are non-normaly distributed, ...
                strongly reccomanded to use non-parametric test')
27         else
28             if CountNONNormaly<(columns/2)
29                 display('Some features are non-normaly distributed, ...
                    strongly reccomanded to use non-parametric test ...
                    or apply log-transformation')
30             end
31         end
32     end
33 end
34 Condition=(CountNONNormaly≠0);
35 if Condition
36     prompt = 'Do you want to log-transform your data? Enter yes if ...
        you do or no if you do not: ';
37     str = input(prompt,'s');
38     switch str
39         case 'yes'
40             for i=1:columns
41                 DATA(:,i)=log( DATA(:,i));
42             end
43             CountNONNormaly=0;

```

```

44         Condition=(CountNONNormaly==0);
45     case 'no'
46         CountNONNormaly≠0;
47         Condition=(CountNONNormaly==0);
48     otherwise
49         display('error, please enter yes or no')
50     end
51 end
52 %% Relevance Analysis
53 [TAB1, Pvalues]=Stat (DATA,VarNames',Condition);
54 PosSignChanging=(Pvalues<0.05);
55 VarNamesSignificant=VarNames (PosSignChanging);
56 SignificantDATA=DATA (:,PosSignChanging);
57 %% Correlation
58 D=correlation (SignificantDATA, Condition);
59 %% Find the maximum number of features
60 m=MaxNumberOfFeatures (DATA);
61 %% Find the best combination of relevant and non-redudant features
62 BestGoodCombos=Redundancy (D.Mask,m);
63 [d l]=size (BestGoodCombos)
64
65 for j=1:d
66     for i=1:l
67         BestComb{i}=VarNamesSignificant (BestGoodCombos{j,i});
68     end
69 end
70
71
72 end

1 function [TAB1, Pvalues]=Stat (DATA,VarNames,condition)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function computes the statistical analysis for the input ...
4     matrix and
5     %%the p-value between two different conditions. As input, it takes the
6     %%matrix with observations as rows and features (predictors) as columns,
7     %%the states orlabels (in binary values) as last column.
8     %%The second input, condition, will help understand if parametric or
9     %%not parametric analysis needs to be performed. Parametric test is
10    %%performed using t-test whereas non-parametric test is performed using
11    %%Wilcoxon rank test.
12    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13    %%
14    [rows columns]=size (DATA);

```

```

15 %Find positions of the two different conditions
16 Pos_Features_Experiment=find(DATA(:,end));
17 Pos_Features_Rest=find(DATA(:,end)==0);
18
19 if ~condition % if the condition is false; positive condition is ...
    that the data are normally distributed
20     for i=1:(columns-1)
21         [p(i),h(i)]=ranksum...
22             (DATA(Pos_Features_Experiment,i),...
23             DATA(Pos_Features_Rest,i));
24     end
25     median_Experiment=median(DATA(Pos_Features_Experiment,...
26         1:(columns-1))');
27     SD_Experiment=std(DATA(Pos_Features_Experiment,...
28         1:(columns-1))');
29     Per_Experiment=prctile(DATA(Pos_Features_Experiment,...
30         1:(columns-1)),[25 50 75])';
31
32     median_Rest=median(DATA(Pos_Features_Rest,1:(columns-1))');
33     SD_Rest=std(DATA(Pos_Features_Rest,1:(columns-1))');
34     Per_Rest=prctile(DATA(Pos_Features_Rest,1:(columns-1)),[25 50 ...
35         75])';
36
37     Pvalues=p(1:(columns-1))';
38
39     TAB1=table(VarNames, median_Rest, SD_Rest, Per_Rest, ...
40         median_Experiment, SD_Experiment, Per_Experiment, Pvalues);
41
42 else % if the condition is true
43     for i=1:(columns-1)
44         [h(i),p(i)]= ...
45             ttest(DATA(Pos_Features_Experiment,i),DATA(Pos_Features_Rest,i));
46     end
47
48 %%
49 %Generate Table
50 mean_Experiment=mean(DATA(Pos_Features_Experiment,...
51     1:(columns-1))');
52 SD_Experiment=std(DATA(Pos_Features_Experiment,...
53     1:(columns-1))');
54 Per_Experiment=prctile(DATA(Pos_Features_Experiment,...
55     1:(columns-1)),[25 50 75])';
56
57 mean_Rest=mean(DATA(Pos_Features_Rest,...
58     1:(columns-1))');
59 SD_Rest=std(DATA(Pos_Features_Rest,...
60     1:(columns-1))');

```

```

57     Per_Rest=prctile(DATA(Pos_Features_Rest,...
58         1:(columns-1)),[25 50 75]');
59
60     Pvalues=p(1:(columns-1))';
61
62     TAB1=table(VarNames,mean_Rest, SD_Rest, Per_Rest, ...
63         mean_Experiment, SD_Experiment, Per_Experiment, Pvalues);
64 end
65
66 end

```

```

1 function [D]=correlation(DATA, condition)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function gerenates correlation matrices. The
4 %%condition is needed to understand if parametric or non-parametric
5 %%correlation analysis needs to be performed.The output is a structure
6 %%with the rho values and p-values of the diagonal of the correlation
7 %%matrices.
8 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9 %%
10 [rows columns]=size(DATA)
11
12 %Not parametric correlation
13 if ~condition
14
15     [c_E ...
16         p_E]=corr(DATA(:,1:columns-1),DATA(:,1:columns-1),'Type','Spearman');
17     Buffer_E=((abs(c_E)>0.7).*(p_E<0.05)); % Condition to be highly ...
18         correlated and significant. The threshold of 0.7
19         %can be changed to a more restricted one
20     Buffer_E(logical(eye(size(Buffer_E)))) = 0;
21     D=struct('Mask',Buffer_E, 'rhoValues',c_E,'pValues',p_E);
22 else %Parametric correlation
23
24     [c_E ...
25         p_E]=corr(DATA(:,1:columns-1),DATA(:,1:columns-1),'Type','Pearson');
26     Buffer_E=(abs(c_E)>0.7).*(p_E<0.05);
27     Buffer_E(logical(eye(size(Buffer_E)))) = 0;
28     D=struct('Mask',Buffer_E, 'rhoValues',c_E,'pValues',p_E);
29 end

```

```

1 function n=MaxNumberOfFeatures(DATA)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function calculates the maximum number of features that the model
4 %%can contain in order to avoid overfitting
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7 [rows columns]=size(DATA);
8 Pos_Features_Experiment=find(DATA(:,end));
9 Pos_Features_Rest=find(DATA(:,end)==0);
10 s=size(Pos_Features_Experiment);
11 if ~isequal(s,size(Pos_Features_Rest));
12     error('same number of instances during the two conditions');
13 end
14 NumbofSub=length(Pos_Features_Experiment);
15 n=NumbofSub/10; %Rule of Thumb
16 n=ceil(n);
17 if n>columns
18     n=columns;
19 else
20     n=n;
21 end

```

```

1 function BestGoodCombosPos=Redundancy(M,m)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function generates all the possible combinations of features. As
4 %%input, it takes the correlation matrix (M) and the max number of ...
5 %%that a combination can contain.
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 n=length(M)
8 GoodCombosPos=[];
9 k=1;
10 X=1;
11 if m>n
12     m=n;
13 else
14     m=m;
15 end
16
17 for k=1:m
18     combos = nchoosek((1:n),k);
19     Ncombos=size(combos,1);
20     for i=1:Ncombos
21
22         Sums(i)=sum(sum(M(combos(i,:),combos(i,:))));

```

```

23
24     end
25
26     s(k).GoodCombosPos=find(Sums==0);
27     s(k).combos=combos;
28     Condition=(length(s(k).GoodCombosPos)>0);
29     if Condition
30
31         BestGoodCombosPos{k}=s(k).GoodCombosPos;
32         BestGoodCombos{k}=s(k).combos(s(k).GoodCombosPos,:);
33
34     end
35     clear GoodCombosPos combos Sums
36 end
37
38 end

1 function [trainingData] = generateTablestraining(BestCombPos, ...
    datasettraining)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function generates the training datasets for each features ...
    combination
4 %%It takes as input the data from Folder 2 and the features' ...
    combinations computed in
5 %%FeatureSelectionProcess function.
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 datasettraining=datasettraining(:, 2:end); %only HRV features and class
8 VarNames=datasettraining.Properties.VariableNames
9 D=table2array(datasettraining);
10 [l p]=size(BestCombPos);
11 for j=1:p
12     [n m]=size(BestCombPos{:,j});
13     for i=1:n
14         b=BestCombPos{:,j}
15         trainingData{j,i}=[datasettraining(:,b(i,:)), datasettraining(:,end)]
16     end
17 end
18 end

1 function N=MinNumbN(DATA)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function computes the minimum number of iteration for unbalanced
4 %%dataset. However, the user can decide if increase the minimum ...

```

```

        number of
5 %%iteration (N).
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 DATA=table2array(DATA);
8 Pos_Features_Experiment=find(DATA(:,end)==1);
9 Pos_Features_Control=find(DATA(:,end)==0);
10 n=length(Pos_Features_Control);
11 fprintf( 'The minimun nuber of repition is estimated: %d.\n', n)
12 prompt ='Do you prefer an higher number? Answer yes or no....'
13 str = input(prompt,'s');
14 switch str
15     case 'yes'
16         number='insert here your number of repetition....'
17         str1 = input(number);
18
19         N=str1
20     case 'no'
21         N=n;
22 end
23 end

1 function [trainedClassifier, validationAccuracy, CP, AUC] = ...
    trainClassifierTree(trainingData, N)
2 % trainClassifier(trainingData)
3 % returns a trained classifier and its accuracy.
4 % This code recreates the classification model trained in
5 % Classification Learner app.
6 %
7 % Input:
8 %     trainingData: the training data of same data type as imported
9 %     in the app (table or matrix).
10 %
11 % Output:
12 %     trainedClassifier: a struct containing the trained classifier.
13 %     The struct contains various fields with information about the
14 %     trained classifier.
15 %
16 %     trainedClassifier.predictFcn: a function to make predictions
17 %     on new data. It takes an input of the same form as this training
18 %     code (table or matrix) and returns predictions for the response.
19 %     If you supply a matrix, include only the predictors columns (or
20 %     rows).
21 %
22 %     validationAccuracy: a double containing the accuracy in
23 %     percent. In the app, the History list displays this

```

```

24 %         overall accuracy score for each model.
25 %
26 % Use the code to train the model with new data.
27 % To retrain your classifier, call the function from the command line
28 % with your original data or new data as the input argument ...
    trainingData.
29 %
30 % For example, to retrain a classifier trained with the original ...
    data set
31 % T, enter:
32 %     [trainedClassifier, validationAccuracy] = trainClassifier(T)
33 %
34 % To make predictions with the returned 'trainedClassifier' on new ...
    data T,
35 % use
36 %     yfit = trainedClassifier.predictFcn(T)
37 %
38 % To automate training the same classifier with new data, or to ...
    learn how
39 % to programmatically train classifiers, examine the generated code.
40
41
42
43 % Extract predictors and response
44 % This code processes the data into the right shape for training the
45 % classifier.
46
47 [m n]=size(trainingData);
48 for j=1:m
49     [k n(j)]=size(find(~cellfun(@isempty,trainingData(j,:))));
50     for i=1:n(j)
51
52         inputTable = trainingData{j,i};
53
54         predictorNames = inputTable.Properties.VariableNames(1:end-1);
55         predictors = inputTable(:,(predictorNames));
56         response = inputTable(:,end);
57         response=array2table(response);
58         isCategoricalPredictor = false(1, ...
            length(predictorNames(1:end-1)));
59         k=MaxNumbCrossVal(inputTable);
60
61         for f=1:N
62             indices = crossvalind('Kfold',table2array(response),k);
63             for l = 1:k
64                 test = (indices == l); train = ~test;

```

```

65         end
66
67         % Train a classifier
68         % This code specifies all the classifier options and ...
           trains the classifier.
69         VariableDescriptions = ...
70             hyperparameters('fitcensemble',...
71                 predictors(train,:),...
72                 response(train,:), 'Tree');
73         classificationTree = ...
           fitcensemble(predictors(train,:), response(train,:),...
74             'OptimizeHyperparameters', 'auto',...
75             'HyperparameterOptimizationOptions',...
76             struct('AcquisitionFunctionName',...
77                 expected-improvement-plus));
78
79
80         % Create the result struct with predict function
81         predictorExtractionFcn = @(t) t(:, predictorNames);
82         treePredictFcn = @(x) predict(classificationTree, x);
83
84         trainedClassifier(j,i,f).predictFcn = @(x) ...
           treePredictFcn(predictorExtractionFcn(x));
85
86         % Add additional fields to the result struct
87         trainedClassifier(j,i,f).RequiredVariables = ...
           inputTable.Properties.VariableNames;
88         trainedClassifier(j,i,f).ClassificationTree = ...
           classificationTree;
89         trainedClassifier(j,i,f).About = 'This struct is a ...
           trained classifier exported from Classification ...
           Learner R2016a.';
90         trainedClassifier(j,i,f).HowToPredict = sprintf('To make ...
           predictions on a new table, T, use: \n yfit = ...
           c.predictFcn(T) \nreplacing ''c'' with the name of ...
           the variable that is this struct, e.g. ...
           ''trainedClassifier''. \n \nThe table, T, must ...
           contain the variables returned by: \n ...
           c.RequiredVariables \nVariable formats (e.g. ...
           matrix/vector, datatype) must match the original ...
           training data. \nAdditional variables are ignored. \n ...
           \nFor more information, see <a ...
           href="matlab:helpview(fullfile(docroot, ''stats'', ...
           ''stats.map''), ...
           ''appclassification_exportmodeltoworkspace'')">How to ...
           predict using an exported model</a>.');

```

```

91
92
93     %% Validation
94     ValidationData=[predictors(test,:), response(test,:)];
95     [validationAccuracy{j,i,f}, CP{j,i,f}, AUC{j,i,f}, ...
        fig]=testingVAL(ValidationData, trainedClassifier(j,i,f))
96     end
97 end
98 end
99 end
100
101 function [validation, C, AUC, fig]=testingVAL(Testingdata, ...
        trainedClassifier)
102 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
103 %%This function validates the n iteration
104 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
105
106 yfit = trainedClassifier.predictFcn(Testingdata);
107 response=Testingdata(:,end);
108 response=table2array(response);
109 [row column]=size(yfit);
110 correctprediction=zeros(row,1);
111 for i=1:row
112     if yfit (i)==response(i)
113         correctprediction(i)=1;
114     else
115         correctprediction(i)=0;
116     end
117 end
118 validation=sum(correctprediction)/length(correctprediction);
119 C = confusionmat(response,yfit);
120 CP = classperf(response,yfit);
121
122
123 [X,Y,T,AUC,OPTROCPT,SUBY,SUBYNAMES] = perfcurve(response,yfit,1);
124 fig= plot(X,Y,OPTROCPT(1),OPTROCPT(2),'r*'), grid on;
125 xlabel('False positive rate')
126 ylabel('True positive rate')
127 title('ROC')
128
129 end

1 function [trainedClassifier, validationAccuracy, CP, AUC] = ...
        trainClassifierLDA(trainingData, N)
2 % trainClassifier(trainingData)

```

```

3 % returns a trained classifier and its accuracy.
4 % This code recreates the classification model trained in
5 % Classification Learner app.
6 %
7 % Input:
8 %     trainingData: the training data of same data type as imported
9 %     in the app (table or matrix).
10 %
11 % Output:
12 %     trainedClassifier: a struct containing the trained classifier.
13 %     The struct contains various fields with information about the
14 %     trained classifier.
15 %
16 %     trainedClassifier.predictFcn: a function to make predictions
17 %     on new data. It takes an input of the same form as this training
18 %     code (table or matrix) and returns predictions for the response.
19 %     If you supply a matrix, include only the predictors columns (or
20 %     rows).
21 %
22 %     validationAccuracy: a double containing the accuracy in
23 %     percent. In the app, the History list displays this
24 %     overall accuracy score for each model.
25 %
26 % Use the code to train the model with new data.
27 % To retrain your classifier, call the function from the command line
28 % with your original data or new data as the input argument ...
29 %     trainingData.
30 % For example, to retrain a classifier trained with the original ...
31 %     data set
32 %     T, enter:
33 %     [trainedClassifier, validationAccuracy] = trainClassifier(T)
34 % To make predictions with the returned 'trainedClassifier' on new ...
35 %     data T,
36 %     use
37 %     yfit = trainedClassifier.predictFcn(T)
38 %
39 %
40 % Extract predictors and response
41 % This code processes the data into the right shape for training the
42 % classifier.
43 [m n]=size(trainingData);
44 for j=1:m
45     [k n(j)]=size(find(~cellfun(@isempty,trainingData(j,:))));

```

```

46     for i=1:n(j)
47         inputTable = trainingData{j,i};
48
49         predictorNames = inputTable.Properties.VariableNames(1:end-1);
50         predictors = inputTable(:, (predictorNames));
51         response = inputTable(:,end);
52         response=array2table(response);
53         isCategoricalPredictor = false(1, ...
54             length(predictorNames(1:end-1)));
55         k=MaxNumbCrossVal(inputTable);
56         for f=1:N
57             indices = crossvalind('Kfold',table2array(response),k);
58             for l = 1:k
59                 test = (indices == l); train = ~test;
60             end
61
62             % Train a classifier
63             % This code specifies all the classifier options and ...
64             % trains the classifier.
65             classificationDiscriminant = ...
66                 fitcdiscr(predictors(train,:),response(train,:),...
67                     'OptimizeHyperparameters','auto',...
68                     'HyperparameterOptimizationOptions',...
69                     struct('AcquisitionFunctionName',...
70                         'expected-improvement-plus'));
71
72             predictorExtractionFcn = @(t) t(:, predictorNames);
73             discriminantPredictFcn = @(x) ...
74                 predict(classificationDiscriminant, x);
75
76             trainedClassifier(j,i,f).predictFcn = @(x) ...
77                 discriminantPredictFcn(predictorExtractionFcn(x));
78
79             % Add additional fields to the result struct
80             trainedClassifier(j,i,f).RequiredVariables = ...
81                 inputTable.Properties.VariableNames;
82             trainedClassifier(j,i,f).ClassificationDiscriminant = ...
83                 classificationDiscriminant;
84             trainedClassifier(j,i,f).About = 'This struct is a ...
85                 trained classifier exported from Classification ...
86                 Learner R2016a.';
87             trainedClassifier(j,i,f).HowToPredict = sprintf('To make ...
88                 predictions on a new table, T, use: \n yfit = ...
89                 c.predictFcn(T) \nreplacing ''c'' with the name of ...
90                 the variable that is this struct, e.g. ...

```

```

        'trainedClassifier'. \n \nThe table, T, must ...
        contain the variables returned by: \n ...
        c.RequiredVariables \nVariable formats (e.g. ...
        matrix/vector, datatype) must match the original ...
        training data. \nAdditional variables are ignored. \n ...
        \nFor more information, see <a ...
        href="matlab:helpview(fullfile(docroot, 'stats', ...
        'stats.map'), ...
        'appclassification_exportmodeltoworkspace')">How to ...
        predict using an exported model</a>.');
80     %% Validation
81     ValidationData=[predictors(test,:), response(test,:)];
82     [validationAccuracy{j,i,f}, CP{j,i,f}, AUC{j,i,f}, ...
        fig]=testingVAL(ValidationData, trainedClassifier(j,i,f))
83
84
85
86     end
87 end
88 end
89 end
90
91 function [validation, C, AUC, fig]=testingVAL(Testingdata, ...
        trainedClassifier)
92 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
93 %%This function validates the n iteration
94 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
95
96 yfit = trainedClassifier.predictFcn(Testingdata);
97 response=Testingdata(:,end);
98 response=table2array(response);
99 [row column]=size(yfit);
100 correctprediction=zeros(row,1);
101 for i=1:row
102     if yfit (i)==response(i)
103         correctprediction(i)=1;
104     else
105         correctprediction(i)=0;
106     end
107 end
108 validation=sum(correctprediction)/length(correctprediction);
109 C = confusionmat(response,yfit);
110 CP = classperf(response,yfit);
111
112
113 [X,Y,T,AUC,OPTROCPT,SUBY,SUBYNAMES] = perfcure(response,yfit,1);

```

```

114 fig= plot(X,Y,OPTROCPT(1),OPTROCPT(2),'r*'), grid on;
115 xlabel('False positive rate')
116 ylabel('True positive rate')
117 title('ROC')
118
119 end

```

```

1 function [trainedClassifier, validationAccuracy, CP, AUC] = ...
    trainClassifierKNN(trainingData,N)
2 % trainClassifier(trainingData)
3 % returns a trained classifier and its accuracy.
4 % This code recreates the classification model trained in
5 % Classification Learner app.
6 %
7 % Input:
8 %     trainingData: the training data of same data type as imported
9 %     in the app (table or matrix).
10 %
11 % Output:
12 %     trainedClassifier: a struct containing the trained classifier.
13 %     The struct contains various fields with information about the
14 %     trained classifier.
15 %
16 %     trainedClassifier.predictFcn: a function to make predictions
17 %     on new data. It takes an input of the same form as this training
18 %     code (table or matrix) and returns predictions for the response.
19 %     If you supply a matrix, include only the predictors columns (or
20 %     rows).
21 %
22 %     validationAccuracy: a double containing the accuracy in
23 %     percent. In the app, the History list displays this
24 %     overall accuracy score for each model.
25 %
26 % Use the code to train the model with new data.
27 % To retrain your classifier, call the function from the command line
28 % with your original data or new data as the input argument ...
    trainingData.
29 %
30 % For example, to retrain a classifier trained with the original ...
    data set
31 % T, enter:
32 %     [trainedClassifier, validationAccuracy] = trainClassifier(T)
33 %
34 % To make predictions with the returned 'trainedClassifier' on new ...
    data T,

```

```

35 % use
36 %     yfit = trainedClassifier.predictFcn(T)
37 %
38
39
40
41
42 % Extract predictors and response
43 % This code processes the data into the right shape for training the
44 % classifier.
45 [m n]=size(trainingData);
46 for j=1:m
47     [k n(j)]=size(find(~cellfun(@isempty,trainingData(j,:))));
48     for i=1:n(j)
49         inputTable = trainingData{j,i};
50
51         predictorNames = inputTable.Properties.VariableNames(1:end-1);
52         predictors = inputTable(:,(predictorNames));
53         response = inputTable(:,end);
54         response=array2table(response);
55         isCategoricalPredictor = false(1, ...
56             length(predictorNames(1:end-1)));
57         % Train a classifier
58         % This code specifies all the classifier options and trains ...
59         % the classifier.
60         k=MaxNumbCrossVal(inputTable);
61
62         for f=1:N
63             indices = crossvalind('Kfold',table2array(response),k);
64             for l = 1:k
65                 test = (indices == l); train = ~test;
66             end
67             classificationKNN = ...
68                 fitcknn(predictors(train,:),response(train,:),...
69                     'OptimizeHyperparameters','auto',...
70                     'HyperparameterOptimizationOptions',...
71                     struct('AcquisitionFunctionName',...
72                         'expected-improvement-plus'))
73
74             % Create the result struct with predict function
75             predictorExtractionFcn = @(t) t(:, predictorNames);
76             knnPredictFcn = @(x) predict(classificationKNN, x);
77             trainedClassifier(j,i,f).predictFcn = @(x) ...
78                 knnPredictFcn(predictorExtractionFcn(x));
79
80             % Add additional fields to the result struct

```

```

77         trainedClassifier(j,i,f).RequiredVariables = ...
           inputTable.Properties.VariableNames;
78         trainedClassifier(j,i,f).ClassificationKNN = ...
           classificationKNN;
79         trainedClassifier(j,i,f).About = 'This struct is a ...
           trained classifier exported from Classification ...
           Learner R2016a.';
80         trainedClassifier(j,i,f).HowToPredict = sprintf('To make ...
           predictions on a new table, T, use: \n yfit = ...
           c.predictFcn(T) \nreplacing ''c'' with the name of ...
           the variable that is this struct, e.g. ...
           ''trainedClassifier''. \n \nThe table, T, must ...
           contain the variables returned by: \n ...
           c.RequiredVariables \nVariable formats (e.g. ...
           matrix/vector, datatype) must match the original ...
           training data. \nAdditional variables are ignored. \n ...
           \nFor more information, see <a ...
           href="matlab:helpview(fullfile(docroot, ''stats'', ...
           ''stats.map''), ...
           ''appclassification_exportmodeltoworkspace'')">How to ...
           predict using an exported model</a>');
81         %% Validation
82         ValidationData=[predictors(test,:), response(test,:)];
83         [validationAccuracy{j,i,f}, CP{j,i,f}, AUC{j,i,f}, ...
           fig]=testingVAL(ValidationData, trainedClassifier(j,i,f))
84
85
86
87         end
88     end
89 end
90 end
91
92 function [validation, C, AUC, fig]=testingVAL(Testingdata, ...
           trainedClassifier)
93 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
94 %%This function validates the n iteration
95 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96
97 yfit = trainedClassifier.predictFcn(Testingdata);
98 response=Testingdata(:,end);
99 response=table2array(response);
100 [row column]=size(yfit);
101 correctprediction=zeros(row,1);
102 for i=1:row
103     if yfit (i)==response(i)

```

```

104         correctprediction(i)=1;
105     else
106         correctprediction(i)=0;
107     end
108 end
109 validation=sum(correctprediction)/length(correctprediction);
110 C = confusionmat(response,yfit);
111 CP = classperf(response,yfit);
112
113
114 [X,Y,T,AUC,OPTROCPT,SUBY,SUBYNAMES] = perfcurve(response,yfit,1);
115 fig= plot(X,Y,OPTROCPT(1),OPTROCPT(2),'r*'), grid on;
116 xlabel('False positive rate')
117 ylabel('True positive rate')
118 title('ROC')
119
120 end

1 function [trainedClassifierFinal, CPFinal, ValAUCFinal, ...
           BestFeaturesPos]=ClassifierSelection(BestCombPos, AUC, CP, ...
           trainedClassifier)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function selects the best classifier for each machine learning
4 %%methods used.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7 [l g z]= size(AUC)
8 tf = cellfun('isempty',AUC) % true for empty cells
9 AUC(tf) = {0}
10 AUC1=cell2mat(AUC);
11
12 for i=1:g
13     for j=1:z
14         [value(j), row(j)]=max(AUC1(:,i,j));
15         Column(j)=i;
16         index(j)=j;
17     end
18 end
19
20 position=[row', Column', index'];
21 for i=1:row
22     for j=1:Column
23         BestFeaturesPos{i,j}=BestCombPos{ Column(j),row(i)}
24     end
25 end

```

```

26
27 ValAUCFinal=value;
28
29 [g,h]=size(position);
30 for j=1:g;
31     h=1
32     CPFinal{j}=CP{position(j,h), position(j,h+1), position(j,h+2)};
33     trainedClassifierFinal{j}=trainedClassifier(position(j,h),position(j,h+1), ...
        position(j,h+2))
34 end
35 end

1 function [validation, C, AUCavaraged, fig]=testing(Testingdata, ...
    trainedClassifier)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function tests the best classifier on Folder 3 and gives the ...
    accuracy
4 %%performance, confusio matrix, AUC value and the ROC curve.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 [m,n]=size(trainedClassifier);
7 for j=1:m
8     for f=1:n
9         trainedClassifier1=trainedClassifier{j,f}
10        %trainedClassifier.predictFcn =trainedClassifier1.predictFcn;
11 yfit = trainedClassifier1.predictFcn(Testingdata);
12 response=Testingdata(:,end);
13 response=table2array(response);
14 [row column]=size(yfit);
15 correctprediction=zeros(row,1);
16 for i=1:row
17     if yfit (i)==response(i)
18         correctprediction(i)=1;
19     else
20         correctprediction(i)=0;
21     end
22 end
23 validation{j,f}=sum(correctprediction)/length(correctprediction);
24 C {j,f}= confusionmat(response,yfit);
25 CP{j,f} = classperf(response,yfit);
26
27
28 [X,Y,T,AUC{j,f}, OPTROCPT,SUBY,SUBYNAMES] = perfcurve(response,yfit,1);
29 fig= plot(X,Y,OPTROCPT(1),OPTROCPT(2),'r*'), grid on;
30 xlabel('False positive rate')
31 ylabel('True positive rate')

```

```

32 title('ROC')
33 end
34 end
35 AUC=cell2mat(AUC);
36 AUCavaraged=mean(AUC);
37 end

1 function [trainedClassifierFinal, CPFinalModel, AUCT, ...
    BestFeaturesPos]=ModelSelection(NFeaturesALL, CP, AUCTestingAll, ...
    trainedClassifierALL)
2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3 %%This function selects among the different machine learning methods the
4 %%best classifier: max AUC and the lowest number of features used.
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 [l g]= size(AUCTestingAll)
7
8 for i=1:l
9     [value(i),index(i)]= max([AUCTestingAll{: ,i}]);
10    row(i)=i;
11    col(i)=index(i);
12    values(i)=value(i);
13    %position=[row, col, value];
14 end
15
16 [m n]=size(NFeaturesALL);
17 for j=1:m
18     L(j)=length([NFeaturesALL{: ,j}]);
19     [valueL(j),indexL(j)]=min(L(j));
20     rowL(j)=j;
21     colL(j)=indexL(j);
22     valuesL(j)=valueL(j);
23     %position=[row, col, value];
24 end
25 position=[row', col', value'];
26 [AUCFinal,ind]=max(position(:,3));
27 IndexFinal=position(ind,1:end-1);
28
29 positionL=[rowL', colL', valueL'];
30 [L,indL]=min(positionL(:,3));
31 IndexFinalL=positionL(indL,1:end-1);
32 AUCFinalL=AUCTestingAll(:,IndexFinalL(1,2));
33
34
35 CombPos=NFeaturesALL(IndexFinal(1,1),1);
36 CombPosM=cell2mat(CombPos);

```

```
37 BestFeaturesPos=CombPosM(IndexFinal(1,2),:);
38 AUCFinalL=cell2mat(AUCFinalL);
39 %AUCFinal=cell2mat(AUCFinal)
40 if AUCFinalL<=AUCFinal
41     AUCT=AUCFinal;
42     Index=IndexFinal;
43 else
44     AUCT=AUCFinalL
45     Index=IndexFinalL;
46 end
47 trainedClassifierFinal=trainedClassifierALL(Index(1,1),Index(1,2))
48 CPFinalModel=CP(Index(1,1),Index(1,2));
49 end
```

Appendix B

Supplementary Materials

B.1 Bland-Altman plots

The Bland-Altman plots for the six HRV features that showed to be good surrogates of the equivalent short ones are reported here. The Bland-Altman plots were generated for all the time scales (i.e., from 5 min to 30 sec) during rest and stress conditions.

Bland-Altman Plot during Rest

Bland-Altman Plot during Stress

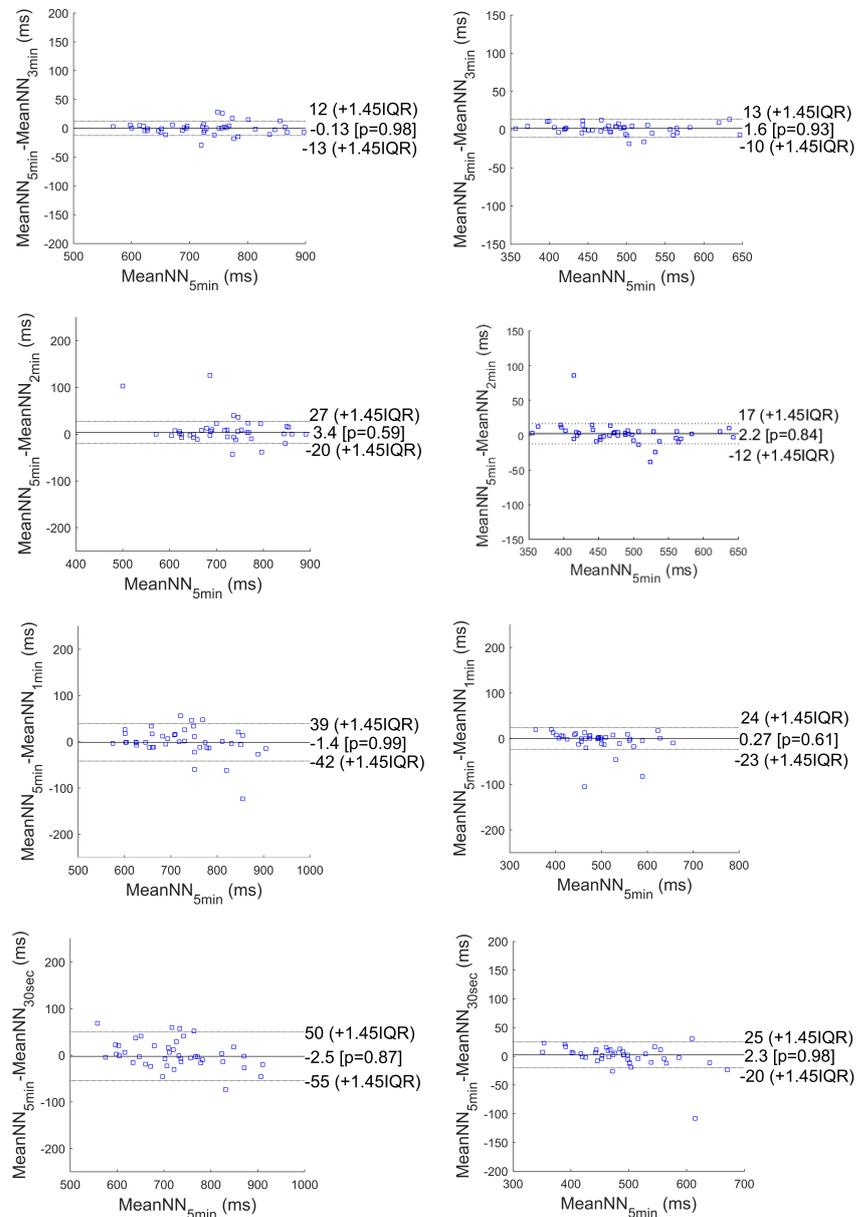


Figure B.1: Bland-Altman plots for MeanNN across time scales during rest and stress conditions. The x-axis is the short HRV feature and the y-axis is the bias of the ultra-short compared to the short HRV feature. The area between the two dotted lines represents the interval between the 95%LoA and the black line represents the bias. The reference line of no bias is $y = 0$. IQR: Interquartile range; p:p-value between the short and ultra-short HRV features (Wilcoxon's rank test)

Bland-Altman Plot during Rest

Bland-Altman Plot during Stress

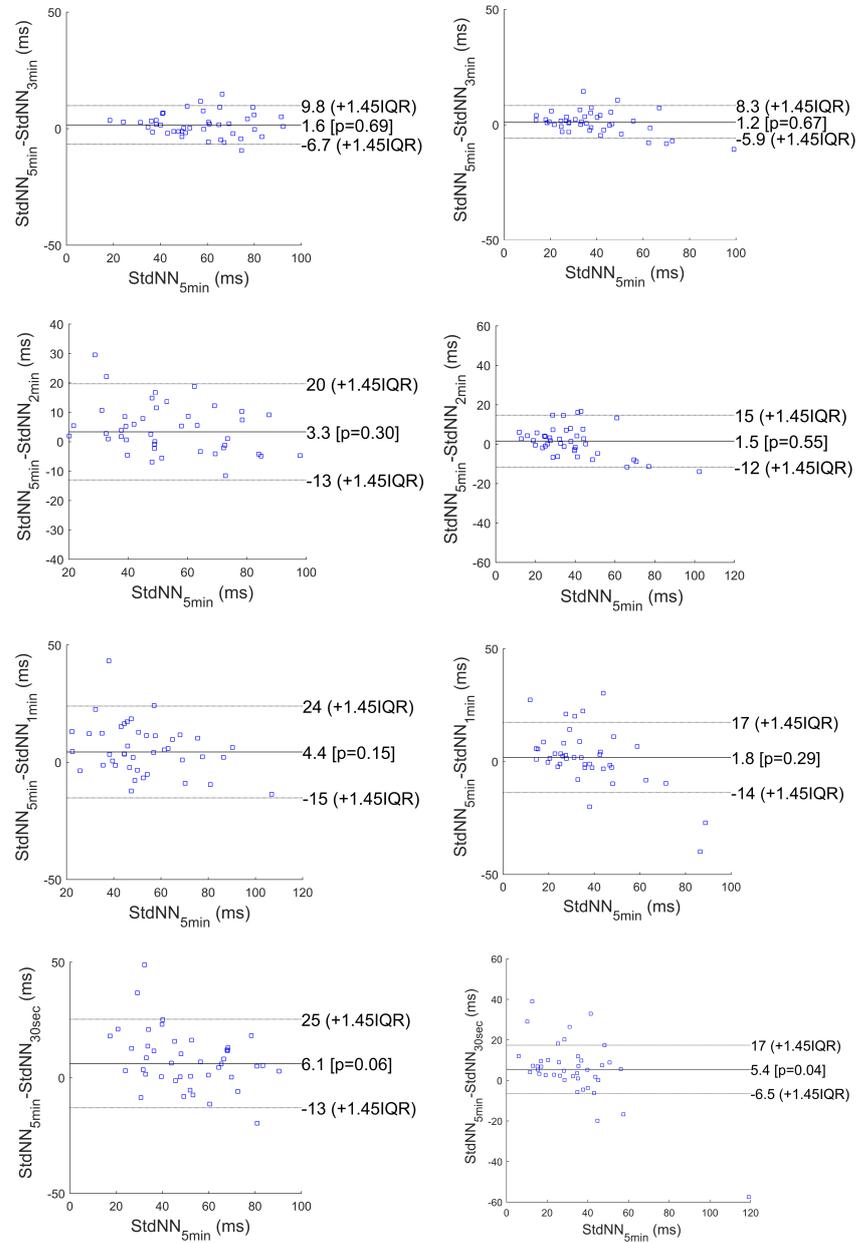


Figure B.2: Bland-Altman plots for StdNN across time scales during rest and stress conditions. The x-axis is the short HRV feature and the y-axis is the bias of the ultra-short compared to the short HRV feature. The area between the two dotted lines represents the interval between the 95%LoA and the black line represents the bias. The reference line of no bias is $y = 0$. IQR: Interquartile range; p:p-value between the short and ultra-short HRV features (Wilcoxon's rank test).

Bland-Altman Plot during Rest

Bland-Altman Plot during Stress

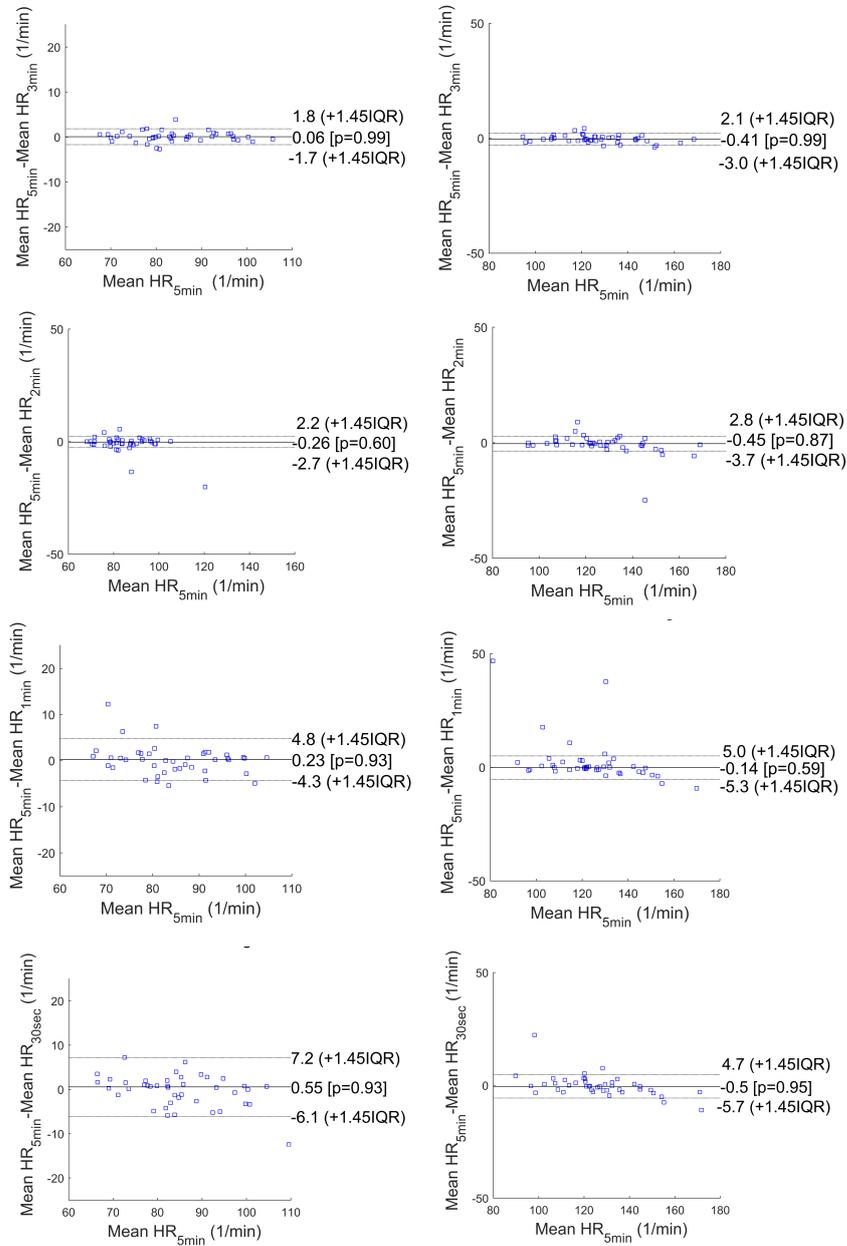


Figure B.3: Bland-Altman plots for MeanHR across time scales during rest and stress conditions. The x-axis is the short HRV feature and the y-axis is the bias of the ultra-short compared to the short HRV feature. The area between the two dotted lines represents the interval between the 95%LoA and the black line represents the bias. The reference line of no bias is $y = 0$. IQR: Interquartile range; p:p-value between the short and ultra-short HRV features (Wilcoxon's rank test).

Bland-Altman Plot during Rest

Bland-Altman Plot during Stress

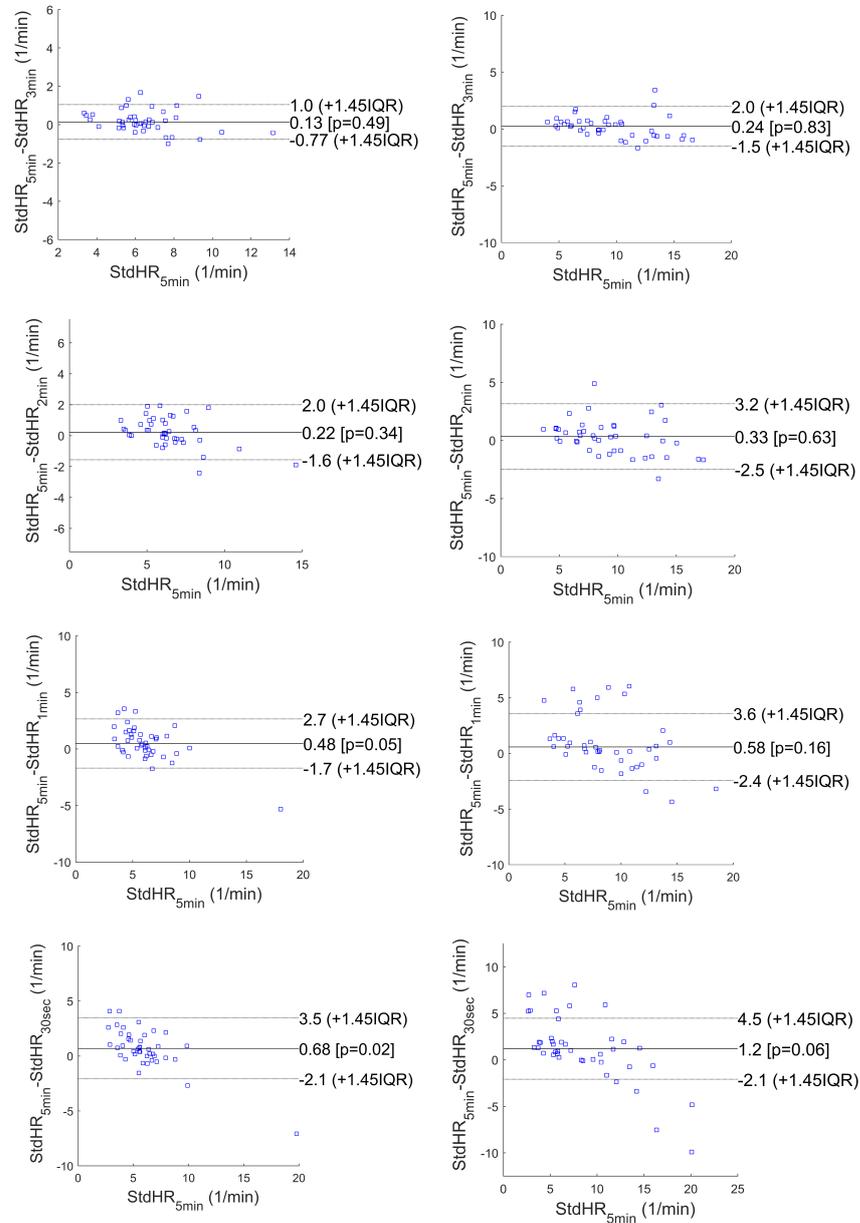


Figure B.4: Bland-Altman plots for StdHR across time scales during rest and stress conditions. The x-axis is the short HRV feature and the y-axis is the bias of the ultra-short compared to the short HRV feature. The area between the two dotted lines represents the interval between the 95%LoA and the black line represents the bias. The reference line of no bias is $y = 0$. IQR: Interquartile range; p:p-value between the short and ultra-short HRV features (Wilcoxon's rank test).

Bland-Altman Plot during Rest

Bland-Altman Plot during Stress

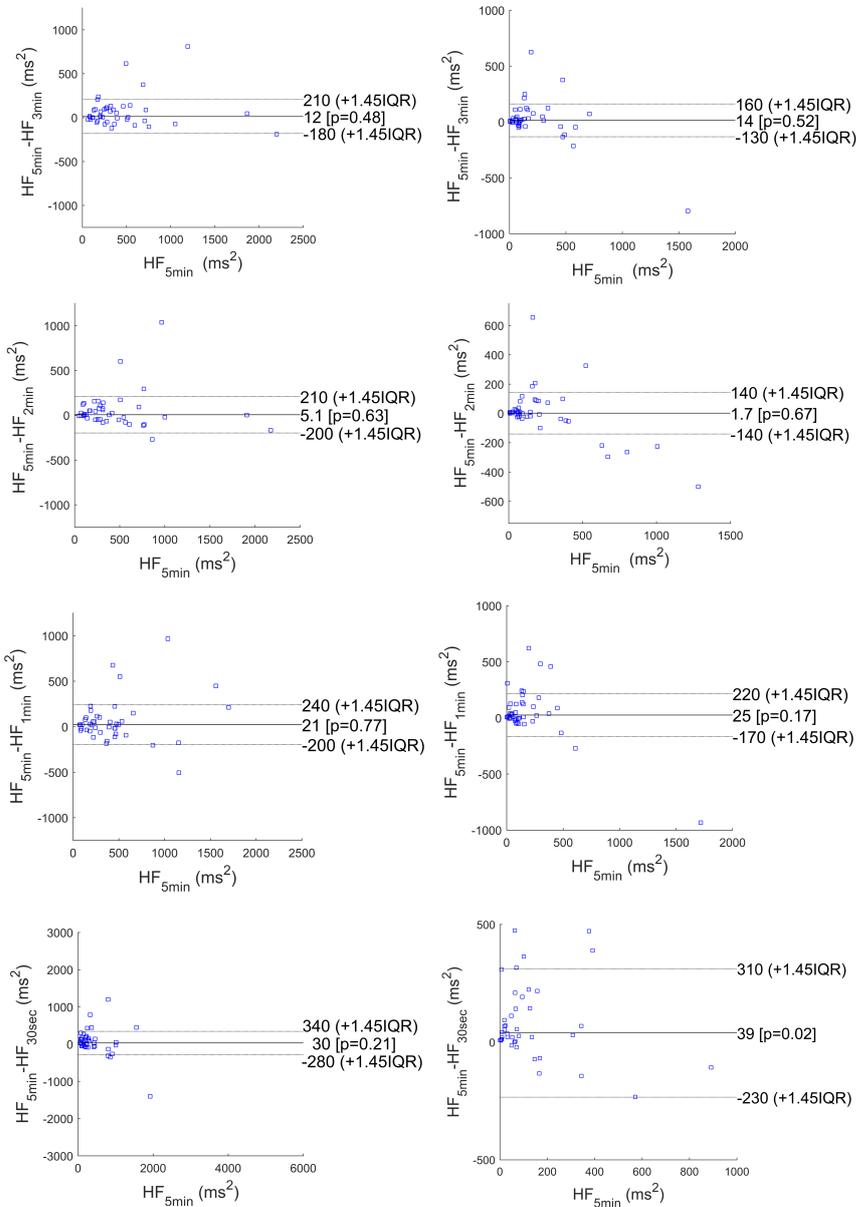


Figure B.5: Bland-Altman plots for HF across time scales during rest and stress conditions. The x-axis is the short HRV feature and the y-axis is the bias of the ultra-short compared to the short HRV feature. The area between the two dotted lines represents the interval between the 95%LoA and the black line represents the bias. The reference line of no bias is $y = 0$. IQR: Interquartile range; p:p-value between the short and ultra-short HRV features (Wilcoxon's rank test).

Bland-Altman Plot during Rest

Bland-Altman Plot during Stress

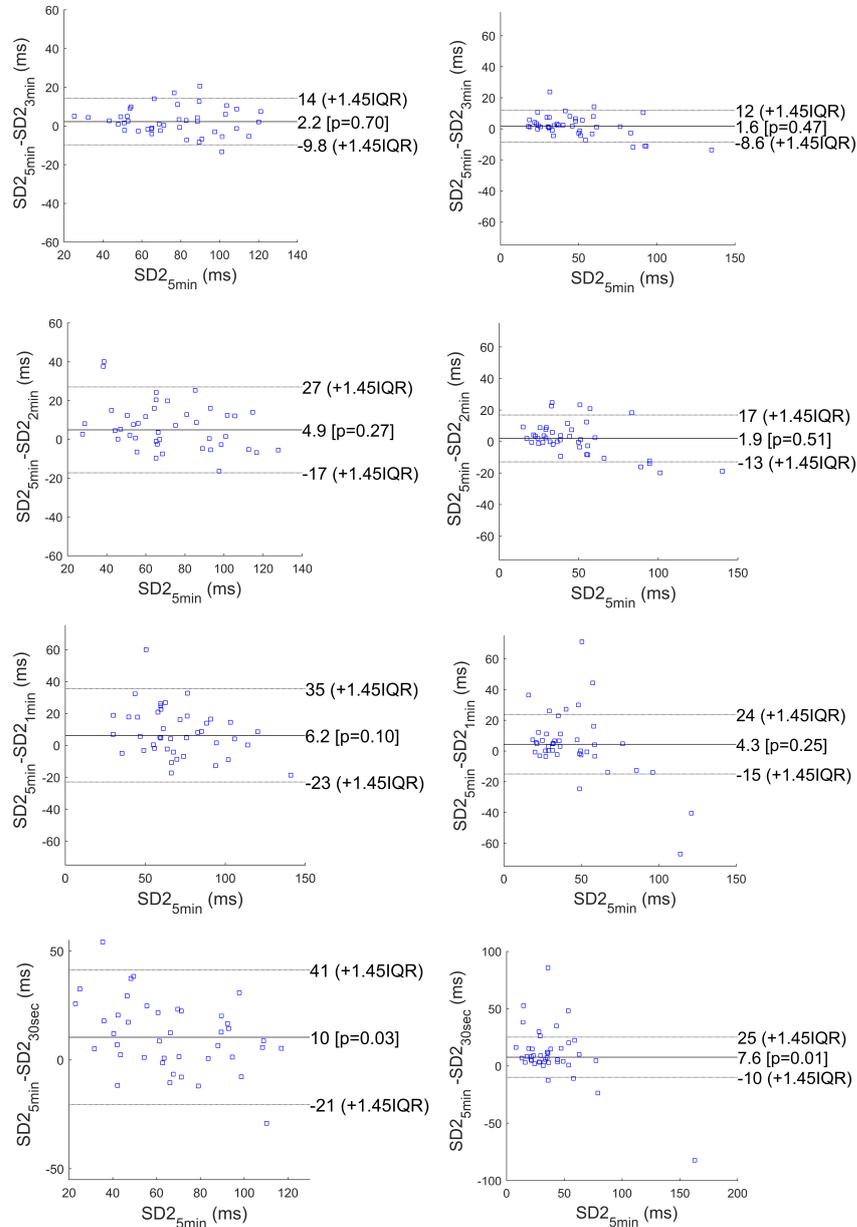


Figure B.6: Bland-Altman plots for SD2 across time scales during rest and stress conditions. The x-axis is the short HRV feature and the y-axis is the bias of the ultra-short compared to the short HRV feature. The area between the two dotted lines represents the interval between the 95%LoA and the black line represents the bias. The reference line of no bias is $y = 0$. IQR: Interquartile range; p:p-value between the short and ultra-short HRV features (Wilcoxon's rank test).

B.2 Questionnaire and ethical approvals

This section includes the questionnaires used during the rest sessions of the SCWT and VGC protocols, and the ethical approval letters for the two experimental studies conducted during my Ph.D. study.

The following questionnaires were used during the rest session of the SCWT and VGC protocols respectively. During the SCWT test, the subjects were invited to talk, while the researcher collected the answers in the prepared sheet. During the VGC, the questionnaire was administrated via PC using SurveyMonkey software.

Sub ID= XXX

Date:.....

Time:.....

Personal Information

1. What is your ID?

2. What is your gender?

Male

Female

3. How old are you?

4. What is your height (in meters)?

5. What is your weight (in kg)?

6. Are you taking any medication at the moment?

Yes

No

7. If yes, can you please list them below?

8. Do you have any health problems that you are aware of?

Yes

No

9. if yes, which ones?

Sleep disorder

Cardiovascular problems

- Stroke
- Chronic fatigue
- Untreated diabetes
- Uncontrolled thyroid disease
- Epilepsy
- Other (please specify):

10. Do you use any drug? (This will be kept strongly confidential)

- Yes
- No

2. If you are a female...

11. Are you pregnant?

- Yes
- No

12. If not, in which day of your menstrual cycle are you in?

Sub ID=XXX

Date:.....

Time:.....

Personal Information

1. What is your name?

2. What is your gender?

Male

Female

3. How old are you?

4. What is your height (in meters)?

5. What is your weight (in kg)?

6. Are you taking any medication at the moment?

Yes

No

7. If yes, can you please list them below?

8. Do you have any health problems that you are aware of?

Yes

No

9. if yes, which ones?

Sleep disorder

Cardiovascular problems

- Stroke
- Chronic fatigue
- Untreated diabetes
- Uncontrolled thyroid disease
- Epilepsy
- Other (please specify):

10. Do you use any drug? (This will be kept strongly confidential)

- Yes
- No

2. If you are a female...

11. Are you pregnant?

- Yes
- No

12. If not, in which day of your menstrual cycle are you in?

3. General Information

13. Are you right-handed?

- Yes
- No

14. Do you have normal or corrected-to-normal vision?

- Yes
- No

15. Are you skilled in using mouse and keyboard?

- Yes
- No

16. How many hours per day do you spend at PC?

17. Do you play video games?

Yes

No

18. If yes, how many hours do you spent per day?

18. Have you ever played to counter-strike game?

Yes

No

19. If yes, would you consider yourself familiar with the game?

14th April 2014

Warwick
Medical School

PRIVATE
Rossana Castaldo
86 Kensington Road
Coventry
CV5 6GH

Dear Rossana,

Study Title and BSREC Reference: *High Mental Stress Detection via Short-Term HRV Analysis* **REGO-2014-656**

Thank you for submitting your revisions to the above-named project to the University of Warwick Biomedical and Scientific Research Ethics Sub-Committee for Chair's Approval.

I am pleased to confirm that I am satisfied that you have met all of the conditions and your application meets the required standard, which means that full approval is granted and your study may commence. I would however strongly recommend that you ask an English language proof reader to check your protocol, PIL and consent forms.

I take this opportunity to wish you success with the study and to remind you any substantial amendments require approval from the committee before they can be made. Please keep a copy of the signed version of this letter with your study documentation.

Yours sincerely,

PP 

David Davies
Chair
Biomedical and Scientific
Research Ethics Sub-Committee

**Biomedical and Scientific
Research Ethics Subcommittee**
A010 Medical School Building
Warwick Medical School,
Coventry, CV4 7AL.
Tel: 02476-151875
Email: BSREC@Warwick.ac.uk

Medical School Building
The University of Warwick
Coventry CV4 7AL United Kingdom
Tel: +44 (0)24 7657 4880
Fax: +44 (0)24 7652 8375

THE UNIVERSITY OF
WARWICK

18th June 2015

Warwick
Medical School

PRIVATE
Dr Leandro Pecchia
Engineering
University of Warwick
Coventry
CV4 7AL

Dear Dr Pecchia,

Study Title and BSREC Reference: *High Mental Stress Detection via Short-Term HRV Analysis* REGO-2014-656 AM01

Thank you for submitting a substantial amendment application for the above-named project to the University of Warwick's Biomedical and Scientific Research Ethics Sub-Committee.

I am pleased to confirm that the changes that you wish to make to this study have been approved.

Please keep a copy of the signed version of this letter with your study documentation.

Yours sincerely



Professor Scott Weich
Chair
Biomedical and Scientific
Research Ethics Sub-Committee

**Biomedical and Scientific
Research Ethics Sub-Committee**
A010 Medical School Building
Warwick Medical School,
Coventry, CV4 7AL.
Tel: 02476-528207
Email: BSREC@warwick.ac.uk

Medical School Building
The University of Warwick
Coventry CV4 7AL United Kingdom
Tel: +44 (0)24 7657 4880
Fax: +44 (0)24 7652 8375

THE UNIVERSITY OF
WARWICK