

A FUSION OF CLASSICAL AND DEEP
MACHINE LEARNING FOR MOBILE HEALTH
AND BEHAVIOR MONITORING WITH
WEARABLE SENSORS

Martin Gjoreski

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Supervisor: Prof. Dr. Matjaž Gams, Jožef Stefan Institute, Ljubljana, Slovenia
Co-Supervisor: Dr. Mitja Luštrek, Jožef Stefan Institute, Ljubljana, Slovenia

Evaluation Board:

Prof. dr. Bogdan Filipič, Chair, Jožef Stefan Institute, Ljubljana, Slovenia
Prof. dr. Mojca Ciglarič, Member, Faculty of Computer and Information Science,
University of Ljubljana, Ljubljana, Slovenia
Prof. dr. Thomas Wittenberg, Member, Fraunhofer Institute for Integrated Circuits,
Erlangen, Germany

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL



Martin Gjoreski

A FUSION OF CLASSICAL AND DEEP MACHINE
LEARNING FOR MOBILE HEALTH AND BEHAVIOR
MONITORING WITH WEARABLE SENSORS
Doctoral Dissertation

SPOJ KLASIČNEGA IN GLOBOKEGA STROJNEGA
UČENJA ZA MOBILNO SPREMLJANJE ZDRAVJA IN
OBNAŠANJA Z NOSLJIVIMI SENZORJI
Doktorska disertacija

Supervisor: Prof. Dr. Matjaž Gams

Co-Supervisor: Dr. Mitja Luštrek

Ljubljana, Slovenia, November 2020

To my parents and my grandparents.

На моите родители и на моите баба и дедо (од двете страни).

Acknowledgments

First of all, I would like to thank my supervisor Prof. Dr. Matjaž Gams and co-supervisor Dr. Mitja Luštrek, who supported and guided me during the overall PhD process at the Jožef Stefan Institute (JSI). I am grateful they have provided me with a proper environment that nurtured and encouraged new scientific ideas. I am also thankful that they have granted me the freedom to explore.

I would like to thank the members of the PhD committee for their efforts, timely evaluation, and for providing constructive comments that improved the quality of the thesis.

I would also like to thank the Slovenian Research Agency who funded my research via the funding for Young Researchers and two bilateral projects, one with N. Macedonia and one with Germany.

The work in this thesis significantly improved thanks to the external collaborators with whom we co-authored joint publications: Dr. Teena Hassan, Dr. Jens-Uwe Garbas, Dr. Veljko Pejović, and Dr. Gregor Poglajen (MD).

Additionally, I would like to thank all my colleagues from the Department of Intelligent Systems at the JSI for enabling relaxed and productive working environment. Especially, I would like to thank Vito Janko, Anton Gradišek, Junoš Lukan, Jani Bizjak, Gašper Slapničar, Borut Budna, Nejc Mlakar and Monika Simjanoska, who had a direct influence on the work presented in this thesis.

Next, I would like to thank my close friends who helped me go through the overall PhD process and made it more enjoyable: Valentin Gjorgjioski, Stevanche Nikoloski, Vladimir Kuzmanovski, Tome Eftimov, Gjorgji Nusev and Dejan Kostadinovski, and my friends from back home.

Finally, this thesis would not have been possible without the closest ones, my girlfriend Maria-Paula, my brother Hristijan and my parents, Roza and Slave. Maria-Paula has the ability to make work-related problems disappear. Hristijan is my close friend, my close collaborator and my ghost supervisor. My parents are my main source of motivation and my life compass. Their hard work and sacrifice made all of this possible.

Abstract

Commercially available smartphones, smart glasses, smartwatches, and smart rings are just a few examples of sensor-packed devices that are enabling the technological revolution currently underway. To further extend the successful applicability of wearable devices in sectors such as mobile health, methods for accurate measurements of psycho-physiological information are required. However, accessing psycho-physiological information using wearable devices remains challenging. One reason is that the relationship between sensor data and human psycho-physiological states is not as unambiguous as the relationship between sensor data and individual physical states is. Thus, we are facing a question: How can we transform wearable sensor data into valuable human health and behavior information? Such information has the potential to improve healthcare, decrease healthcare costs, enhance the automotive industry, enhance sports performance, improve the quality of life and, ultimately, save human lives.

For a decade, deep learning (DL) has dominated the AI world by achieving a breakthrough in several areas such as image processing, natural language processing, and reinforcement learning. Thus, a successful fusion of classical machine learning (ML) and DL methods could lead to beyond state-of-the-art (SOTA) results for mobile health and behavior monitoring.

This thesis proposes a general method for combining expert knowledge, classical ML, and DL to enable the transformation of wearable sensor data into valuable health and behavior information. The method includes: (i) learning from a large-body of expert-defined features using classical ML; (ii) a novel DL architecture, named Spectro-Temporal ResNet, which learns directly from raw sensor data with the potential to discover useful patterns, previously unknown to experts; and (iii) a fusion of classical ML and end-to-end DL approaches, which utilizes meta-learning. The method is applied in seven domains of mobile health and behavior monitoring. These include chronic heart failure detection from heart sounds, locomotion recognition from smartphone sensors, distracted-driving detection from physiological and video-based sensors, stress, emotion, and cognitive-load monitoring from physiological sensors, and blood pressure estimation from ECG sensors. In all these experimental domains, the proposed method achieved better performance than the baseline and the SOTA methods, which were analyzed in the studies.

The combination of a variety of sensors enables better modeling of the human psycho-physiological states as different sensors provide information about different parts of the human nervous system. For example, a sweat sensor can measure the activation of the sympathetic nervous system, and a heart sensor can measure the activation of the sympathetic and the parasympathetic nervous system jointly. Additionally, the combination of a variety of sensors and models enables enhanced robustness to noise in the data. Finally, the proposed meta-learning enables performance optimization, personalization, and can consider temporal dependence in the data, improving the performance of monitoring the human physical, physiological and psychological states.

Povzetek

Komercialno dostopni pametni telefoni, pametna očala, pametne ure in pametni prstani so le nekateri primeri naprav, opremljenih s senzorji, ki omogočajo tehnološko revolucijo, ki jo doživljamo. Za nadaljnjo razširitev uspešne uporabe nosljivih naprav v sektorjih, kot je mobilno zdravje, so potrebne metode za natančno merjenje psiho-fizioloških informacij. Vendar pa je dostop do psiho-fizioloških informacij z uporabo nosljivih naprav še vedno izziv. Eden od razlogov je, da odnos med senzorskimi podatki in človekovimi psiho-fiziološkimi stanji ni tako nazoren, kot je odnos med senzorskimi podatki in posameznimi fizičnimi stanji. Tako se soočamo z vprašanjem: kako naj senzorske podatke pretvorimo v dragocene informacije o zdravju in vedenju ljudi? Dragocene informacije, ki lahko izboljšajo zdravstvo, zmanjšajo stroške zdravstva, izboljšajo avtomobilsko industrijo, izboljšajo športne zmogljivosti, izboljšajo kakovost življenja in na koncu rešijo človeška življenja.

V zadnjem desetletju je globoko učenje (GU) prevladovalo v svetu umetne inteligence, saj je doseglo preboj na več področjih, kot so obdelava slik, obdelava naravnega jezika in spodbujevano učenje. Tako bi lahko uspešen spoj klasičnega strojnega učenja (SU) in metode GU pripeljal do vrhunskih rezultatov za mobilno spremljanje zdravja in obnašanja z nosljivimi senzorji.

Ta disertacija predlaga splošno metodo za združevanje strokovnega znanja, klasičnega SU in GU, da se omogoči pretvorba nosljivih senzorskih podatkov v dragocene podatke o zdravju in obnašanju. Metoda vključuje: (i) učenje iz velikega števila strokovno opredeljenih značilk z uporabo SU; (ii) novo arhitekturo za GU, imenovano Spectro-Temporal ResNet, ki se neposredno uči iz surovih senzorskih podatkov s potencialom za odkrivanje uporabnih vzorcev, neznanih strokovnjakom; in (iii) spoj klasičnega SU in GU, ki uporablja meta-učenje. Metoda je bila uporabljena na sedmih področjih mobilnega spremljanja zdravja in obnašanja. Sem spadajo kronično odkrivanje srčnega popuščanja s pomočjo srčnih zvokov, prepoznavanje premikanja s senzorji pametnih telefonov, odkrivanje motenj vožnje iz fizioloških in video senzorjev, spremljanje stresa, čustev in kognitivne obremenitve s fiziološkimi senzorji ter ocena krvnega tlaka iz senzorjev EKG. Predlagana metoda je na vseh eksperimentalnih področjih dosegala boljše rezultate kot izhodiščne in najsodobnejše metode, ki so bile analizirane v študijah.

Kombinacija različnih senzorjev omogoča boljše modeliranje človekovih psiho-fizioloških stanj, saj različni senzorji dajejo informacije o različnih delih človeškega živčnega sistema. Na primer senzor, ki spremlja potenje, lahko meri aktivacijo simpatičnega živčnega sistema, senzor, ki spremlja srce, pa lahko skupaj izmeri aktivacijo simpatičnega in parasimpatičnega živčnega sistema. Poleg tega kombinacija različnih senzorjev in modelov omogoča boljšo robustnost na šum v podatkih. Predlagano meta-učenje omogoča tudi optimizacijo uspešnosti, personalizacijo in lahko modelira časovno odvisnost podatkov ter izboljša uspešnost spremljanja človekovih fizičnih, fizioloških in psiholoških stanj.

Contents

Acknowledgments	vii
Abstract	ix
Povzetek	xi
Contents	xiii
List of Figures	xv
List of Tables	xvii
List of Algorithms	xix
Abbreviations	xxi
1 Introduction	1
1.1 Challenges	2
1.2 Approach and Hypotheses.....	3
1.3 Scientific Contributions.....	3
1.4 Impact and Publications	3
1.5 Thesis Overview	5
2 Background	7
2.1 Sensors	8
2.2 Mobile Health and Behavior Monitoring.....	10
2.2.1 Sensing Physiological Parameters.....	11
2.2.2 Sensing Psychological Parameters	11
2.2.3 Sensing Behavioral Parameters	14
2.3 Machine Learning	14
2.3.1 Classical Machine Learning.....	15
2.3.2 Deep Learning	16
2.3.3 Fusion in Machine Learning.....	16
3 Method for Mobile Health and Behavior Monitoring	17
3.1 Sensor Data.....	17
3.2 Preprocessing.....	18
3.2.1 Data Synchronization	18
3.2.2 Filtering	19
3.2.3 Transformations and Virtual Sensors	23
3.2.4 Normalization.....	25
3.2.5 Windowing.....	28

3.3	Feature Extraction.....	29
3.3.1	Statistical Features.....	29
3.3.2	Frequency Features.....	29
3.3.3	Heart-Related Features.....	29
3.3.4	Galvanic Skin Response Features.....	30
3.4	Feature Selection.....	32
3.5	Classical Machine Learning.....	33
3.6	End-to-End Deep Learning for Sensor Data.....	35
3.7	Fusion of Models and Datasets.....	36
4	Early Work	39
4.1	Monitoring Stress Using Context.....	39
4.2	Non-Invasive Blood Pressure Estimation from ECG.....	41
4.3	Emotion Recognition from Physiological Signals.....	42
4.4	Cognitive-Load Inference from Physiological Signals.....	44
4.5	Summary and Discussion.....	46
5	The Fusion of Classical and Deep Machine Learning	49
5.1	Chronic Heart Failure Monitoring.....	50
5.2	Driver Distractions Monitoring.....	62
5.3	Locomotion Recognition.....	76
5.4	Summary and Discussion.....	92
6	Conclusions	95
6.1	Summary of Contributions.....	97
6.2	Limitations and Future Directions.....	98
6.2.1	Threats to Validity.....	98
6.2.2	From Supervised to Self-Supervised and Unsupervised Learning.....	98
6.2.3	Explainable and Responsible AI.....	98
6.2.4	Cost of AI Systems.....	99
Appendix A		101
	Statistical Features.....	101
	HRV Features.....	101
	Galvanic Skin Response Features (GSR Features).....	101
References		105
Bibliography		121
	Publications Related to the Thesis.....	121
	Journal Articles.....	121
	Conference Papers.....	121
	Book Chapters.....	123
	Workshop Papers.....	123
	Other Publications.....	124
Biography		125

List of Figures

Figure 2.1: From sensor data to health and behavior knowledge.	8
Figure 2.2: Tradeoff between obtrusiveness and noise in wearable sensors [104].	10
Figure 2.3: Fight-or-flight response.	12
Figure 2.4: Classical machine learning (a) vs. (b) end-to-end deep learning [116].	15
Figure 3.1: The architecture of the method for extracting human health and behavior information from wearable sensor data.	17
Figure 3.2: Labeled smartphone sensor data from the SHL dataset.	19
Figure 3.3: Filtering example. Detrended ECG signal.	20
Figure 3.4: Filtering example. Detrended BVP signal.	20
Figure 3.5: Example raw and filtered GSR signal. Used filter: moving average.	21
Figure 3.6: Raw and filtered acceleration signal. Used filter: low-pass.	22
Figure 3.7: Raw (left) and filtered (right) acceleration signal. The color represents the amplitude of a particular frequency at a particular time. Used filter: low-pass.	22
Figure 3.8: Example R-R intervals extracted from ECG (left) and BVP (right) signals.	23
Figure 3.9: Decomposition of a complex signal into a sum of simple sinusoidal signals.	24
Figure 3.10: From time (red graph) to frequency (blue graph) domain.	24
Figure 3.11: Data normalization (raw data).	25
Figure 3.12: Personalized vs. overall min-max normalization.	26
Figure 3.13: Personalized vs. overall standardization.	26
Figure 3.14: Personalized vs. overall robust normalization.	27
Figure 3.15: Personalized vs. overall robust winsorization.	27
Figure 3.16: Personalized vs. overall quantile-based normalization (using the Gaussian probability density function).	28
Figure 3.17: R-R signal in the time domain (left). Spectral representation (normalized Lomb-Scargle periodogram) of the same R-R signal (right).	30
Figure 3.18: Example R-R intervals extracted from PCG signal (phonocardiogram).	30
Figure 3.19: Skin Conductance Response.	31
Figure 3.20: GSR signal with example Skin Conductance Responses (SCRs) detected.	31
Figure 3.21: The combination of two non-informative features can be informative.	33
Figure 3.22: Two examples from an end-to-end CNN model for arousal recognition from the GSR signal. The top graphs represent the normalized GSR input.	36
Figure 4.1: Context-based stress detection.	40
Figure 4.2: Evaluation results for the laboratory stress detector. LOSO evaluation.	40
Figure 4.3: Method for non-invasive blood pressure estimation from ECG.	42
Figure 4.4: Meta-learning for emotion recognition from physiological signals.	44
Figure 4.5: Multi-task learning vs. single-task learning.	45

List of Tables

Table 2.1: Wearable devices equipped with sensors.....	9
Table 4.1: Experimental data for monitoring stress.....	39
Table 4.2: Confusion matrices and evaluation metrics (Precision, Recall and F1 score) for No-Context vs. With-Context stress detection.....	41
Table 4.3: Mean absolute error for cuff-less blood pressure estimation.....	42
Table 4.4: Experimental datasets for the study on emotion recognition.....	43
Table 4.5: Accuracy for dataset-specific models for different modalities (R-R intervals, Galvanic Skin Response and the combination of the two).....	43
Table 4.6: Accuracy (%) for emotion recognition from physiological sensors.....	44
Table 4.7: Accuracy (%) for different modalities (R-R intervals, Galvanic Skin Response, Skin Temperature and the combination of the three) on the Cog. Load dataset.....	44
Table 4.8: Cognitive-load inference from physiological signals. Accuracy for best-performing algorithm (bold) vs. single-task-learning (STL) vs. multi-task learning (MTL) for two experimental datasets, Snake and Cog. Load.....	45
Table 4.9: Method summary. Presenting which components of the proposed general method (Figure 3.1) were used in the early-work studies.....	46
Table 4.10: Results summary. The numbers in the table represent the evaluation score used in each study (Accuracy, F1-score, or Mean Absolute Error).....	47
Table 5.1: Method summary. Presenting which components of the proposed general method (Figure 3.1) were used in the studies that combined ML and DL approaches.....	92
Table 5.2: Results summary. The numbers in the table represent the evaluation score used in each study (Accuracy or F1-score).....	93
Table 6.1: Experimental domains and their relation to the thesis hypotheses.....	96
Table A.1: Features summary.....	103

List of Algorithms

Algorithm 3.1:	Linear detrending using a sliding window.....	21
Algorithm 3.2:	Low-pass filter.	22
Algorithm 3.3:	Peak detection algorithm.	24
Algorithm 3.4:	Feature selection algorithm.....	34
Algorithm 3.5:	Algorithm for hyperparameter tuning.....	35

Abbreviations

AdaBoost	Adaptive Boosting
AI	Artificial Intelligence
AR	Activity Recognition
AU	Action Unit
BP	Blood Pressure
BR	Breathing Rate
BVP	Blood Volume Pulse
CD	Cognitive Drive
CDF	Cumulative Distribution Function
CHF	Chronic Heart Failure
CNN	Convolutional Neural Network
DBP	Diastolic Blood Pressure
DL	Deep Learning
ECG	Electrocardiogram
ED	Emotional Drive
EDA	Electrodermal Activity
EEG	Electroencephalogram
eHealth	Electronic Health
EMO	Emotional Activation
EOG	Electrooculogram
EU	European Union
F1-s	F1-Scores of the Session Classifiers
FC	Fully Connected
FDL	Failure-Loaded Drive
FDN	Failure-Normal Drive
FFT	Fast Fourier transformation
GSR	Galvanic Skin Response
HAR	Human Activity Recognition
HCI	Human Computer Interaction
HMM	Hidden Markov Model
HF	Heart Failure
HR	Heart Rate
HRV	Heart Rate Variability
IBI	Intra-bit-interval
ICT	Information and Communication Technologies
ID3	Iterative Dichotomiser 3
IMU	Inertial Measurement Unit
kNN	k-Nearest-Neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
LSTM	Long Short-Term Memory

MAE	Mean Absolute Error
MAP	Mean Arterial Pressure
mHealth	Mobile Health
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology database
MTL	Multi-Task Learning
NASA-TLX	NASA Task Load Index
NB	Naïve Bayes
ND	Normal Drive
nEDA	Nasal Electrodermal Activity
NN	Neural Network
PAT	Pulse Arrival Time
PCA	Principal Component Analysis
pEDA	Palm Electrodermal Activity
PPG	Photoplethysmogram
PSD	Power Spectral Density
PTT	Pulse Transit Time
PWV	Pulse-Wave Velocity
RD	Relaxed Drive
RF	Random Forest
SBP	Systolic Blood Pressure
SCR	Skin Conductance Response
SD	Sensorimotor Drive
SHL	Sussex-Huawei Locomotion
STL	Single-Task Learning
STResNet	Spectro-Temporal Residual Network
SVM	Support Vector Machine
WHO	World Health Organization
XGB	Extreme Gradient Boosting

Chapter 1

Introduction

In the European Union (EU), 1.7 million persons aged less than 75 years died in 2013. Among them, nearly 34% of such deaths could have been avoided with better healthcare.¹ Similarly, in 2010, the World Health Organization (WHO) accredited 63% of deaths to non-communicable diseases that are largely preventable [1]. These numbers demonstrate that there is significant room for improvement in personal healthcare in the EU. The scientific field that has the potential to influence future personal healthcare considerably is Mobile Health (mHealth), which refers to the usage of mobile, wearable, and wireless communication technologies to improve healthcare delivery, outcomes, research, and data collection [2][3].

For example, fifteen million Europeans suffer from heart failure (HF) [4], which results in an estimated annual cost of 2% of the annual healthcare budget in the EU [5]. Additionally, HF is the most common cause of hospitalization among the over-65 age group [6]. As the EU population continues to age, i.e., the average lifespan continuously increases, more people will be at risk of HF, placing even more substantial burden on healthcare systems. Remote solutions that allow early detection of HF would allow a treating physician to adjust the patient's medical management promptly and thus avoid health complications and the need for hospitalization. Another example includes the one hundred million Europeans who are currently experiencing mental health problems, including anxiety disorders, depression, drug/alcohol use, and bipolar disorders. The mental-health costs amount to €600 billion, which is more than 4% of the EU's GDP. Remote solutions for timely detection and improved management of mental disorders can positively impact the lives of those one hundred million people and can reduce the healthcare costs.

In addition to the medical application of wearable technologies, another domain that can benefit from their application is that of human behavior monitoring. Human behavior is defined by the actions and mannerisms made by individuals in conjunction with themselves or their environment [7]. For example, on the EU roads, 25,000 lives are lost annually, with a vast majority of the accidents caused by human error, often by a distracted driver. Solutions that detect, measure, and analyze human affective states (e.g., driving attention, emotions, stress, cognitive load, sleepiness, etc.), can decrease the loss of human lives and can contribute to the EU's long-term goal of zero fatalities and severe injuries by 2050.²

On the bright side, the proliferation of wearable devices equipped with a variety of sensors allows monitoring human behavior and health. Commercially available smartphones, smart glasses, smartwatches, smart shirts, smart rings, fitness trackers, and dedicated physiological sensors are just a few of such sensor-packed devices. Even video-

¹ Amenable and preventable deaths statistics: <http://ec.europa.eu/eurostat/statistics>

² https://ec.europa.eu/commission/presscorner/detail/en/IP_19_1793

and sound-recording devices can serve as sensors. These devices, in combination with advanced machine learning (ML) methods, can bring significant improvements to our lives. While wearable devices have already achieved commercial success, several issues hamper their potential to improve our everyday lives: (i) lack of labeled data for ML; (ii) lack of ML methods for monitoring subtle physiological changes; (iii) lack of ML methods for monitoring psychological states.

How can we transform wearable sensor data into valuable human health and behavior information? Such information has the potential to improve healthcare, decrease healthcare costs, enhance the automotive industry, enhance sports performance, improve the quality of life, and ultimately save human lives.

1.1 Challenges

The availability of small, wearable, low-cost sensors combined with advanced signal processing and information extraction is driving the revolution in the domain of mobile behavior monitoring. This technological drive enables novel approaches and technologies for accurate measurements in sports analytics, wellbeing, and lifestyle monitoring [8].

To further extend the applicability of wearable sensors in sectors such as mobile health systems and the automotive industry, methods for accurately extracting subtle psychological and physiological information from the sensors are required. However, accessing psycho-physiological information in everyday life remains challenging [9]. One reason is that the relationship between the wearable sensor data and the human psycho-physiological states is not as explicit as is the relation between the wearable sensor data and human physical states. For example, smartphones can count steps and recognize human physical activities (e.g., running vs. walking), but cannot recognize emotions and related affective states (e.g., cognitive load). Even humans cannot recognize psycho-physiological states perfectly. For example, the average accuracy achieved by humans in emotion recognition from speech is 86% [10].

Thus, we are faced with a challenging problem that contains uncertainty at several levels: uncertainty in the ground truth, e.g., the labels that human annotators provide for specific situations may differ; uncertainty in the sensor data, i.e., some wearable devices do not provide reliable sensor data continuously because the wearable-sensor revolution is quite recent and the sensing systems are not mature enough; and, uncertainty in the methods, i.e., for many psycho-physiological tasks, it is not evident which sensor combination with which processing methods would produce the most reliable results. On top of that, labeled datasets are limited. In some cases, they do not even exist because collecting and labeling human- and health-related data is expensive, and it often requires the expertise of medical specialists.

For a decade, deep learning (DL) has dominated the AI world by achieving breakthroughs in image processing [11], [12], [13], [14], natural language processing [15], [16], [17], and reinforcement learning [18], [19]. Conversely to these domains, DL methods for sensor data are quite scarce, and the appropriate DL methods and architectures have yet to be discovered [20]. Thus, studying state-of-the-art machine-learning methods and the possibility of using end-to-end deep learning on wearable sensor data is an open research question. A successful fusion of classical and DL methods could lead to significant advances for mobile health and behavior monitoring.

1.2 Approach and Hypotheses

This thesis developed a general method that combines expert knowledge, classical machine learning (ML), and deep learning (DL) to enable the transformation of the wearable sensor data into valuable health and behavior information. The method includes: (i) learning from a large-body of expert-defined features extracted from the processed data using classical ML; (ii) end-to-end DL that learns directly from the sensor data with the potential to discover useful patterns in the data, previously unknown to experts; and (iii) a fusion of classical ML and end-to-end DL approaches, which utilizes meta-learning.

The main hypotheses that are proposed, investigated, and confirmed in this thesis are as follows:

Hypothesis 1: Combining data from a variety of wearable sensors improves the performance of monitoring human physical, physiological and psychological states using machine learning.

Hypothesis 2: Our method for combining machine-learning models, including end-to-end deep learning, enables an additional meta-learning layer that can combine models to maximize performance, can learn personalized models and can learn models that take into account the temporal dependence in the data, which improves the performance even further for monitoring human physical, physiological and psychological states, compared to the existing state-of-the-art methods.

1.3 Scientific Contributions

This thesis generated the following original contributions:

1. A novel general method that combines expert knowledge, classical machine learning (ML), and deep learning (DL) for extracting human physical, physiological and psychological information from wearable sensor data. The novelty of the method relies on the fusion of classical ML and end-to-end DL approaches, which combines models to maximize performance, can learn personalized models, and/or can learn models that consider the temporal dependence in the data.
2. A novel DL architecture for end-to-end learning specialized for multimodal sensor data.
3. Unified application of the method on seven problem domains: chronic heart failure detection from heart sounds, locomotion recognition from smartphone sensors, driving distractions detection from physiological and video-based sensors, stress, emotion and cognitive-load monitoring from physiological sensors, and blood pressure estimation from ECG sensors.
4. Four new labelled datasets for mobile health and behavior monitoring, and software available for research purposes.

1.4 Impact and Publications

The work in this thesis resulted in the development of ML systems that are used in two European projects: Fit4Work [21] and STRAW [22]. In the Fit4Work project, a context-based stress monitoring module was developed to monitor the mental fitness of older workers using a wrist device equipped with physiological and inertial sensors [23]. The module combined ML models for stress and activity recognition (AR) [24] to improve the accuracy in real-life scenarios. In the second project, the context-based monitoring served

as a basis for the development of an improved ML system that will be used to investigate how day-to-day stress experiences relate to events and activities in a natural office environment.

Additionally, the methods developed in this thesis were incorporated in JSI-Deep [25][26], a method that combines classical ML and DL models for AR from smartphone sensors. JSI-Deep won the international competition for AR, SHL 2018,³ which was a part of the international conference UbiComp 2018. The SHL dataset [28] is the largest dataset for AR collected in real-life settings, which indicates that JSI-Deep is applicable in real life. The experience gained during SHL 2018 was used to win three more international ML competitions, all of them organized as part of A-ranked conferences (UbiComp and IJCNN): SHL 2019,⁴ EMTEQ-AR 2019⁵ [32] and Challenge-UP⁶ 2019 [32]. SHL 2019 was similar to SHL 2018, with one additional hurdle: the competitors had to develop models for different body locations (e.g., hand-worn and pocket-worn smartphones). EMTEQ-AR 2019 was an AR challenge based on inertial measurements (acceleration and gyroscope sensors) derived from head-worn devices. In the Challenge-UP competition, the participants developed ML pipelines using sensor data from wearable sensors, ambient sensors, and vision devices. The competitive success of the ML systems also received significant media coverage and appeared in several national papers and TV news in Slovenia,⁷ N. Macedonia,⁸ and the UK.⁹

A number of publications underlie this thesis: nine journal articles (eight in journals included in a science citation index – SCI), 16 conference papers, two book chapters and nine workshop papers. We started our research by developing methods for monitoring stress using wrist-worn devices equipped with physiological and inertial sensors [23][24]. The analysis showed that stress can be recognized in laboratory settings, with decent accuracy, using standard ML approaches. However, more advanced ML approaches were needed to recognize stress in real life. For that reason, we developed the context-based ML models. The research on monitoring stress resulted in a master’s thesis and two journal publications. One publication in the journal *Sensors* [24], presenting the AR used by the context-based approach; and another publication in the *Journal of Biomedical Informatics*, presenting the overall context-based stress recognition [23]. Encouraged by the satisfactory results achieved by combining several ML classifiers, we continued developing combined approaches for other similar domains. We explored the possibility for cuff-less blood pressure estimation from ECG sensors using a stack of ML classifiers [33][34]. The best performance was achieved by a meta-learning scheme that combined the predictions of several different ML classifiers and enabled person-specific calibration, which resulted in another publication in the journal *Sensors*. Additionally, we started exploring mechanisms for combining datasets, and not just ML classifiers. For that reason, in collaboration with the University of Ljubljana, we performed a study on cognitive load monitoring from sensor data by combining two different datasets using multi-task learning [35][36][37][38]. We also performed a study for combining AR datasets using transfer-learning [39], and a study for emotion recognition from physiological sensors by combining six different datasets using meta-learning [40].

In parallel, we explored end-to-end DL with the potential to discover patterns in the data previously unknown to experts. Our analysis showed that existing DL architectures were not suitable for sensor data. For that reason, we developed the Spectro-Temporal

³ <http://www.shl-dataset.org/activity-recognition-challenge/>

⁴ <http://www.shl-dataset.org/activity-recognition-challenge-2019/>

⁵ <https://github.com/simon2706/Emteq-ARC2019>

⁶ <https://sites.google.com/up.edu.mx/challenge-up-2019/>

⁷ <https://www.delo.si/novice/znanoteh/mimo-gnec-na-cestah-s-slovenskim-algoritmom-112330.html>

⁸ <https://lokalno.mk/algoritam-za-prepoznavane-aktivnosti-razvien-na-feit-pobedi-na-megunaroden-natprevar/>

⁹ <http://www.sussex.ac.uk/broadcast/read/46392>

Residual Network (STResNet), a novel DL architecture specialized in multimodal sensor data. STResNet was successfully applied to four different domains: AR from smartphone sensors [25]; CHF detection from heart sounds [41]; blood pressure estimation from photoplethysmogram (PPG) data [42]; and, monitoring driving distractions from physiological and visual sensors [43]. The study on AR was published in the journal *Information Fusion*. The study on monitoring driving distractions was performed in collaboration with the Fraunhofer Institute of Integrated Circuits, Erlangen, Germany and a joint publication was published in *IEEE Access*. The study on CHF detection was performed in collaboration with the University Medical Center in Ljubljana. The models developed in that study were evaluated on data from real CHF patients in Slovenia. The most advanced version of the system, which combines the STResNet and classical ML using meta-learning, was published in *IEEE Access*. This system is currently being tested on data from new patients, and the results are encouraging. This opens a possibility for the work presented in this thesis to have a real impact on the healthcare in Slovenia.

1.5 Thesis Overview

The remainder of the thesis is organized as follows.

Chapter 2 starts by describing the background in the field of mHealth and behavior monitoring, which is followed by a description of the sensors typically used in related studies. Next, approaches that analyze psychological, physiological and behavioral parameters are described. Finally, the ML approaches for sensor data are presented, including classical ML, DL, and their fusion.

Chapter 3 presents the proposed method, including sections that contain detailed explanations for each component of the method: sensor data, preprocessing, feature extraction, feature selection, classical ML, DL, and the fusion of models and data. Variations of this general method are later used in seven experimental domains.

Chapter 4 presents a summary of the early work performed in four different domains with focus on classical ML: stress recognition from physiological sensors, blood pressure estimation from ECG sensors, emotion recognition from physiological sensors and cognitive-load recognition from physiological sensors. Each subsection includes an introduction, a description of the data, a description of the method, and an overview of the experimental results. In the final subsection, a general summary and discussion are presented.

Chapter 5 presents three additional studies on combining classical ML and DL: chronic heart failure monitoring from heart sounds, driver distractions monitoring from physiological and video-based sensors, and locomotion recognition from smartphone sensors. The three studies are presented via three corresponding journal publications, each of them containing: introduction, related-work analysis, data description, method description, experimental results, discussion and conclusion. Additionally, in the final subsection, a general summary and discussion are presented.

Finally, **Chapter 6** concludes the thesis and provides directions for future work.

Chapter 2

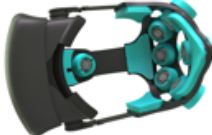
Background

The mHealth field has emerged as a sub-segment of the electronic health (eHealth) field, i.e., a field where information and communication technologies (ICTs) are used for health services and information management [44]. Compared to eHealth, mHealth is focused on the use of mobile, wearable, and wireless communication technologies [1]. Nowadays, computer and mobile applications are available for many types of useful health-related functions such as information management, time management, health-record maintenance and access, communications and consulting, information gathering, clinical decision-making, and medical education and training [46][47]. There is also a growing number of mobile applications for specific behavioral and mental health purposes. For example, applications are available to help with the management of health-related behaviors such as diet, exercise, sleep, smoking, relaxation, and medication adherence. Applications are also available to directly assist with the treatment of mental-health conditions, such as anxiety disorders, depression, eating disorders, psychosis, and for suicide prevention [45][46]. Most of these applications are based on an explicit user input and questionnaires.

The availability of small, wearable, low-cost sensors combined with advanced signal processing and information extraction enable novel approaches and measurements in mHealth, sports analytics, fitness, wellbeing, and lifestyle monitoring [7]. However, since this wearable sensor revolution is quite recent, it also brings many challenges yet to be addressed. Figure 2.1 presents a typical approach for mHealth and behavior monitoring using wearable sensors. The approach includes: wearable sensors to sense information about the user; medical, ICT and domain experts to define the problem, to help with data labeling and to provide guidelines for the modeling process in the specific domain. Next is the ML part, through which the expert knowledge is refined into ML models. Finally, the ML models are used for translating the wearable sensor data into health and behavior knowledge. The extracted knowledge can be useful both for the users and for the experts. For example, the combination of wearable sensors and ML allows for the development of novel approaches for monitoring human physiological (e.g., monitoring patients' heart activity in real life), psychological (e.g., monitoring stress and emotions) and behavioral parameters (e.g., monitoring physical activities). Furthermore, the combination of these parameters enables systems for monitoring mental and physical health in real time.

The next sections are organized as follows: first, the sensing technologies used in mHealth and behavior monitoring studies are described; next, approaches for mHealth and behavior monitoring in the domains related to this thesis are presented; and finally, the ML approaches for sensor data are presented, including classical ML, deep learning and their fusion.

Table 2.1: Wearable devices equipped with sensors.

Device	Description
	Dedicated sensors: ECG sensor, EMG sensor, accelerometer, temperature sensor, gyroscope sensor, magnetometer, etc.)
	Wrist devices: PPG, GSR, HRV, skin temperature, accelerometer
	Smartphone: accelerometer, ambient light sensor, ambient temperature sensor, air humidity sensor, barometer sensor, gyroscope sensor, magnetometer, proximity sensor, pedometer sensor and thermal camera
	Head-worn EEG devices
	Smart ring: PPG, HRV, accelerometers
	Digital stethoscope: phonocardiogram (heart sounds)
	Virtual Reality devices – EEG, EMG, PPG, acceleration sensor, gyroscope gravity sensor and magnetometer

Using thermal cameras, they were able to extract information about the participants' sweating level, which was also found to relate to physiological stress. Similarly, heart rate and heart-related changes (e.g., HRV and blood volume pulse) can be remotely sensed using image analysis [111][112]. For example, recent studies used thermal imaging to remotely estimate body temperature as part of a system for monitoring virus outbreak (e.g., Covid-19) [113].

While there is a variety of wearable devices that can be used in HCI studies, researchers should be aware of the trade-off between sensor obtrusiveness and the noise in the data. The sources of the noise are movement and sensor misplacement. Figure 2.2 presents a comparison between a more obtrusive “Normal ECG” (top graph), secured with wires and electrodes, and a “Contactless ECG” (middle graph). The graph at the bottom presents the output of 3-axis acceleration sensors. The green points represent good sensor data and the red points represent bad sensor data. From the figure it can be seen that the noise in the Normal ECG sensor is much lower compared to the contactless ECG sensor. It can also be seen that even the normal ECG has some noise in the data. This noise can be mitigated to a certain degree by using the data from the acceleration sensors to detect segments. Therefore, the methods used in HCI studies have to be able to deal with multimodal sensor data.

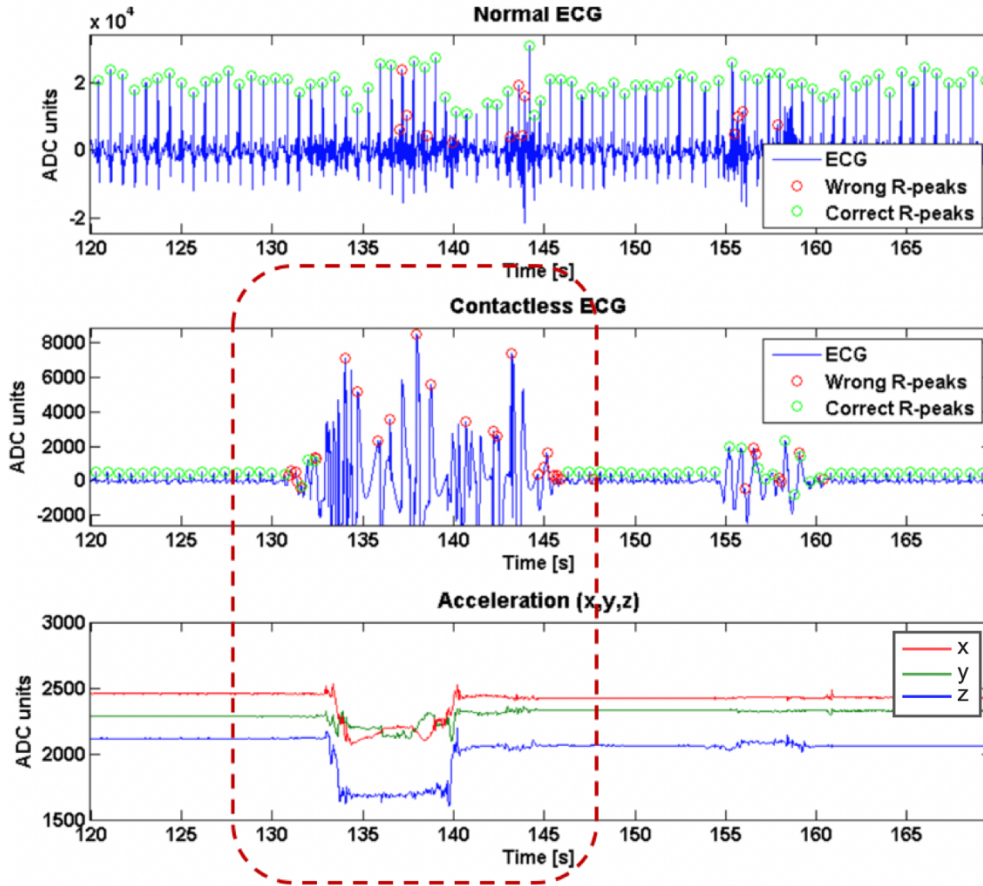


Figure 2.2. Tradeoff between obtrusiveness and noise in wearable sensors [104].

2.2 Mobile Health and Behavior Monitoring

In the StudentLife study [52][53], the authors used automatic sensing data from smartphones to assess students' mental health, activity level, sociability, academic performance, and behavioral trends. Their approach is based on correlation analysis between different aspects of the students' life. One of the main conclusions of that study is: "Results from the StudentLife study show a number of significant correlations between the automatic objective sensor data from smartphones and mental health and educational outcomes of the student body". Ishio and Abe [50] presented a well-being study on aging population from a depopulating village in Japan. They performed correlation analysis on subjective responses and heart rate variability (HRV) parameters, calculated from physiological data gathered via a wrist device. Their findings showed that mental well-being can be modeled using manual answers and physiological data, and that there is a weak correlation between the affect scores of the manual answers and the scores calculated by HRV analysis. Rabbi et al. [51] presented a multimodal mobile sensing system that can simultaneously assess mental and physical health in a population of older adults using a custom device worn on the belt. The system continuously captures motion and refined audio data. From the captured data, they were able to derive automated measurements of physical and mental well-being which correlated well with standardized questionnaires for assessing physical and mental well-being.

Although these studies are based on correlation analysis, they clearly demonstrated the possibility to use mobile and wearable sensors to monitor a variety of mHealth and behavior parameters, including psychological, physiological and behavioral patterns.

2.2.1 Sensing Physiological Parameters

Physiological parameters that are frequently monitored in HCI studies are those that can be measured using wearable physiological sensors. These include: ECG, EEG, BVP, PPG, GSR (see Abbreviations), etc. There are also studies that use virtual sensors. For example, in one of our studies, we worked on cuff-less blood pressure (BP) estimation from ECG sensors. BP is defined as the force of the blood pushing against the walls of the arteries as the heart pumps blood [54] and is measured in millimeters of mercury – mmHg. The maximum pressure during one heart beat is the systolic BP and the minimum pressure in between two heart beats is diastolic BP. Regarding the related work on cuff-less BP estimation, most of the studies use a combination of ECG and PPG sensors [55][56], making the problem even more complex and obtrusive as it requires two devices. The usual techniques used for obtaining the BP rely on Pulse-Wave Velocity (PWV), Pulse Arrival Time (PAT) and Pulse Transit Time (PTT) [57][58][59], all of which require an accurate PPG measurement, which is not simple to get unobtrusively, and yet no clear proof on their relation to the BP is provided [60][61]. Considering the methods that use ECG sensors, Chan et al., [62] and Ahmad et al. [63] presented such studies, however, both of these methods use an additional sensor besides the ECG sensor, i.e., the PPG sensor. The relationship between ECG and BP was previously discussed and tested in few studies [64][65]; however, the results confirm no strong relation between the hypertension occurrence and the morphological changes in the ECG signals.

Based on the physiological parameters, human health status can be monitored. For example, in the study on CHF monitoring, we developed ML models for distinguishing CHF patients vs. healthy people, based on the data from a digital stethoscope which provided heart-sound data. Furthermore, for the CHF patients, we were able to distinguish different CHF stages (e.g., before vs. after hospitalization).

2.2.2 Sensing Psychological Parameters

Sensing psychological parameters is much more challenging compared to sensing physiological parameters. This is because the psychological parameters cannot be directly measured, but they have to be inferred from their physiological manifestation instead.

Definition 2.1 (Stress). “Stress is considered a process by which a stimulus elicits an emotional, behavioral and/or physiological response, which is conditioned by an individual’s personal, biological and cultural context” [67].

Stress is a naturally occurring process in our body, but when present continuously, it can trigger chronic stress. Chronic stress has negative health consequences, such as raised blood pressure, bad sleep, decreased performance and slower body recovery processes [66]. For example, when humans undergo a vigorous event (e.g., training, meeting, exam, etc.), the body is faced with a large physical and psychological stressor which invokes a response of the sympathetic nervous system to meet the increased metabolic demands [68]. The sympathetic nervous system speeds up certain processes within the body (“fight-or-flight” response [69]): it raises the heart rate, sweating rate, blood pressure, etc., some of which can be detected with wearable sensors. The data from wearable sensors allows for developing of stress monitoring systems using signal processing and ML techniques. There are many studies for stress detection using machine learning. The majority of these studies are on stress detection in constrained environments (e.g., laboratory, office, car, etc.) [70], and very few are for stress detection in unconstrained environments [72]. The pioneers in

the field of stress detection using physiological sensors and ML are Healey and Picard [70] who in 2005 presented a quite accurate stress detection system, achieving 97% accuracy while tested in a constrained real-life scenario, while the subjects were driving a car. The downside of this approach is the number of the physiological sensors used, namely: ECG, EMG, GSR and respiration. Even though the presented system is obtrusive, it confirmed that stress detection was possible in a real-life scenario even in 2005. Sarker et al. [72] proposed a time series pattern mining method to detect significant stress episodes. Their stress detection system provides time series of discontinuous and rapidly varying stress data on which they apply the proposed time series pattern mining method. They applied the proposed method to 4 weeks of physiological, GPS, and activity data collected from 38 users in real life. One interesting finding they showed is that the duration of a prior stress episode predicts the duration of the next stress episode. Hovsepian et al. [73] proposed cStress, a method for continuous stress assessment in real life, which was also evaluated in another real-life study [72]. They proved that stress can be detected using a chest belt which provides respiration and ECG data.

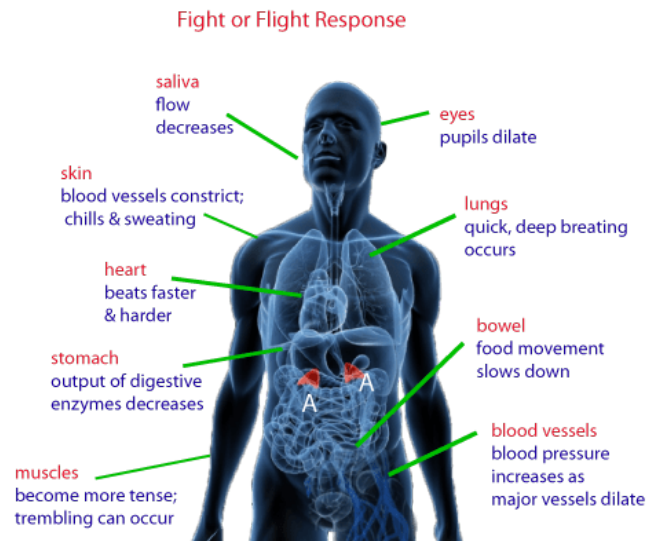


Figure 2.3: Fight-or-flight response.¹⁰

Definition 2.2 (Cognitive load). “Cognitive load is a multidimensional construct representing the load that performing a particular task imposes on the learner’s cognitive system” [74].

Cognitive load is tightly related to stress as the cognitive load in fact represents one type of stressor which activates the fight-or-flight response. Thus, similarly to stress, cognitive load can be inferred using wearable sensors, signal processing and ML techniques. Cognitive load represents an important aspect that modulates human behavior. A timely and reliable assessment of a person’s cognitive load can enable a range of new and improved applications in areas like game-based learning, simulator-based training and considerate HCI. However, the concept remains intangible and is difficult to measure as it depends on the task, the participant, and the interaction between the two. For example, tasks may be objectively more or less demanding and people can have different cognitive capacities. Recent advancements in sensing technology enable less intrusive forms of vital-sign

¹⁰ <https://analyzingtv.wordpress.com/2016/02/29/deterioration-of-the-fight-or-flight-response/>

monitoring and get us closer towards unobtrusive cognitive load inference [75]. For instance, Novak et al. used a wristband to infer cognitive load in a simulated driving environment [77]. The authors argue that cheap wearables may provide enough information about physiological signals to enable binary (“engaged in a task” vs. “not engaged in a task”) classification of the cognitive load, yet are unlikely suitable for inferring the actual level of cognitive load. Schaule et al. used the same wristbands and an N-back task to elicit different levels of cognitive load among office workers [78]. Haapalainen et al. [79] achieved an average accuracy of 80% for binary classification problem (“easy” vs. “difficult” tasks) using personalized ML models and a combination of heat flux and ECG features derived from specialized sensor equipment.

Definition 2.3 (Emotions). Emotions are a complex state that results in psychological and physiological changes that influence our behaving and thinking [94].

When analyzing human emotions, two different approaches can be used, a discrete approach or a continuous approach. In the discrete approach, the emotions are represented as discrete and distinct states, i.e., anger, fear, sadness, happiness, boredom, disgust and neutral [92]. In the continuous approach, the emotions are represented in 2D or 3D space of activeness, valance and dominance [93]. In the emotion recognition from physiological sensors, the main assumption is that there are objectively measurable physiological cues that can be used for predicting the human emotional state. For example, the emotional state of fear usually initiates rapid heartbeat, rapid breathing, sweating and muscle tension. These physiological changes can be captured by wearable devices. Similarly to stress detection studies, there are many studies confirming that emotion recognition can be successful in controlled environments using physiological data or speech [80]. In addition, emotion recognition can be performed using video/image analysis [81] [82] [83]. In general, the methods based on EEG data outperform methods based on other data [84][85], probably due to the fact the EEG provides a more direct channel to one’s mind. Regarding the typical ML approaches for affect recognition, Iacoviello et al. combined discrete wavelet transformation, principal component analysis and support vector machine (SVM) to build a hybrid classification framework using EEG [86]. Khezri et al. used EEG combined with GSR to recognize six basic emotions via K-nearest neighbors (KNN) classifiers [87]. Mehmood and Lee used independent component analysis to extract emotional indicators from EEG, EMG, GSR and ECG [88]. Mikuckas et al. [89] presented an HCI system for emotion recognition that uses spectro-temporal analysis only on R-R intervals, i.e., the time interval between two consecutive heartbeats. More specifically, they focused on recognizing stressful states by means of the heart rate variability (HRV) analysis. Yin et al. [90] presented an approach for emotion recognition from physiological signals using an ensemble of autoencoders. As input, they used features from seven physiological sensors: EEG, GSR, EOG, EMG, ECG, BVP and respiration-rate sensor.

Based on the psychological factors, human mental health can be monitored [95]. For example, Pitting et al. [96] studied mental disorders (panic, social anxiety, obsessive-compulsive, and generalized anxiety disorders) in relation to HRV parameters extracted from ECG data. Cella et al. [97] analyzed physiological measures (HRV and GSR), functioning (activity level), symptoms and medication levels in people with schizophrenia. Tazawa et al. [98] developed ML models to screen for depression and assess severity.

2.2.3 Sensing Behavioral Parameters

Diet, exercise, sleep, smoking cessation, relaxation, and medication adherence are just a few examples of human behavior that can be tracked using wearable devices. For example, there are numerous approaches for sleep monitoring, e.g., sleep monitoring based on physiological signals, sleep monitoring based on accelerometers, sleep monitoring-based smart mattress, etc. [91]. In this thesis, we experimented with human activity recognition (HAR) and monitoring human behavior while driving.

Physical activity is known to be related to well-being [99], sleep quality [104] and overall health [106]. Moderate to vigorous physical activity is associated with decreased risk factors for obesity, cardiovascular and pulmonary diseases, cancer, depression, and increased bone health [99]. Therefore, automatic recognition of daily activities may assist with proper management of pathologies such as obesity, diabetes and cardiovascular diseases [100]. Accurate measurement of physical activity is essential for developing intervention strategies and providing rich contextual information which can be used to infer additional useful information [101]. A more technical overview of the related work on HAR is presented in corresponding Section 5.3.

Monitoring human **behavior while driving** is becoming more and more popular as self-driving vehicles equipped with attractive devices may easily divert driver's attention. Example non-driving related tasks, such as dialing and texting on smartphone, reading, writing, and checking route on a map, are some of the tasks that increase safety risks during driving [102][104]. Since diverted attention of the driver influences driving safety, the ability to sense the driver's physical state and activities is crucial [103]. This data can be gathered using wearable sensors or using sensors embedded in the car. The data could include behavioral cues such as facial expressions and gestures, or measure physiological parameters such as heart rate, respiration rate, and GSR. A more technical overview of the related work on monitoring driving distractions is presented in Section 5.2.

2.3 Machine Learning

People are generating enormous amounts of data every day. However, that data is not informative until it is processed properly. The data can be processed and transformed into useful information and knowledge using ML. ML refers to the ability of computers to optimize a certain performance criterion based on previous experience [114]. There are four main ML types [114]:

- Supervised learning – e.g., classification and regression. In the classification, a training set of already classified (labeled) samples (e.g., by an expert or a user) is given, and by analyzing them, certain classification models are learned. Once learned, the models can be used for classifying new samples.
- Unsupervised learning – e.g., clustering. The main idea of clustering is to determine subgroups (clusters) of data that have something in common by using a set of unlabeled samples.
- Association rule learning – in this type of learning, association rules are extracted using past data. An example of one such association rule would be: “How likely is it that a customer who bought product X would also buy product Y?”.
- Reinforcement learning – enables building computer agents to take actions in a given environment in order to maximize a certain cumulative reward. For example, reinforcement learning was used to build computer agents that can play games, e.g., chess and Go [18].

This thesis mostly deals with supervised learning to build models for: chronic heart failure detection from heart sounds, locomotion recognition from smartphone sensors, driving distractions detection from physiological and video-based sensors, and stress, emotion and cognitive load monitoring from physiological sensors. In the study on blood pressure estimation from ECG sensors, a combination of classification and regression models was used. Additionally, we distinguish between classical ML and end-to-end ML (see Figure 2.4). These two learning types are explained in more detail in the following subsections.

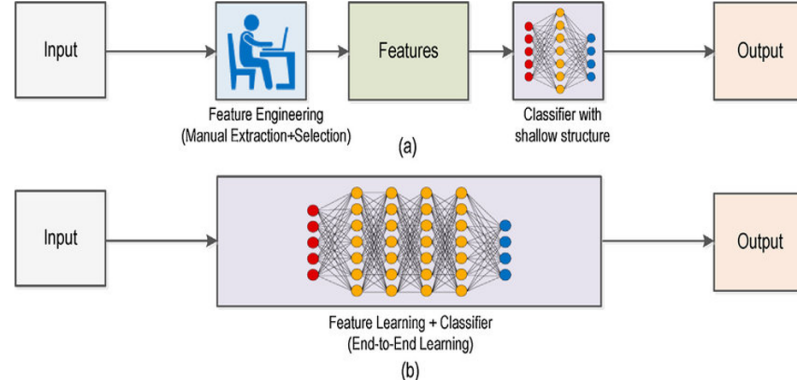


Figure 2.4: Classical machine learning (a) vs. (b) end-to-end deep learning [116].

2.3.1 Classical Machine Learning

In the 1950s, Arthur Samuel developed a program for playing checkers and coined the term “Machine Learning” [117][118]. To minimize the memory requirements of the program, alpha-beta pruning was used, which eventually evolved into the minimax algorithm [118]. The checkers-playing program also had mechanisms to improve its gameplay by memorizing previous positions and analyzing them using a reward function. In the 1960s, the k-Nearest Neighbor algorithm (KNN) was developed, which was the beginning of basic pattern recognition. In the beginning, the algorithm had several versions including: “nearest neighbor pattern classification,” “minimum distance classifier” and “proximity algorithm” [119] [120] [122]. In the 1970s, Quinlan presented the first algorithm for decision tree creation called Iterative Dichotomiser 3 (ID3) [123]. This algorithm was created based on the principles of Occam's razor, with the idea of creating the smallest, most efficient decision tree possible. Quinlan further developed this algorithm with the creation of the C4.5 [124]. After the AI winter in the 1980s, several ground-breaking ML algorithms were invented. Schapire presented Boosting [125], an algorithm that interactively combines several models by “boosting” the weights of misrecognized instances allowing for the models in the ensemble to focus on the misclassified instances. Breiman proposed Bagging (Bootstrap aggregating) [126], an algorithm that trains several classifiers on randomly generated splits of the training sets, and combines their predictions. Dana Cortes and Vladimir Vapnik developed the support vector machines (SVMs) [127] and Ho developed the initial Random Forest [128] that Briemen further improved [129].

Besides the common purpose of these algorithms, i.e., classification, one additional joint characteristic is that they all require features as input. The transformation from sensor data to useful features is another demanding task that needs to be addressed. This transformation process includes data preprocessing, feature extraction and feature selection and offers the possibility to encode expert knowledge into the classification pipeline.

2.3.2 Deep Learning

Contrary to the feature-based classical ML methods, end-to-end methods also exist that can learn directly from the sensor data. Most of these methods are based on deep learning (DL). The most recent DL definition proposed by LeCun¹¹ is as follows:

Definition 2.4 (Deep learning). Deep Learning is building a system by assembling parameterized modules into a (possibly dynamic) computation graph, and training it to perform a task by optimizing the parameters using a gradient-based method [130].

Most DL approaches are based on Artificial Neural Networks (ANNs). The history of ANNs can be traced back to 1943, when Walter Pitts and Warren McCulloch used “threshold logic” to create a model based on the neural networks of the human brain [131]. The basis of backpropagation, i.e., the algorithm used to train DL models by backward propagation of errors, dates from 1960s [132][133][134]. It was significantly improved in 1970 by Linnainmaa [135], and in 1985, it was applied to DL by Rumelhart et al. [136]. In 1979, Fukushima used the first “convolutional neural networks”, Neocognitron [137] which contained multiple pooling and convolutional layers, and in 1989, LeCun et al. [138] applied backpropagation to CNNs on the MNIST dataset. Long short-term memory (LSTM) recurrent neural networks were developed in 1997 [139]. In 2009, ImageNet was created, a publicly available dataset of more than 14 million labeled images [140] that allowed the training of large DL models – e.g., one model can contain more than a million training parameters. In the last decade, the speed of GPUs increased significantly, making it possible to train DL models quite quickly using parallel processing.

A common characteristic of the DL models that are based on CNN and LSTM algorithms is that they can learn directly from the raw data, thus avoiding the need for feature extraction. This characteristic makes end-to-end DL particularly interesting as it holds the potential to discover new useful patterns in the data, previously unknown to experts.

2.3.3 Fusion in Machine Learning

A general data fusion model proposed by JDL Data Fusion Group includes several fusion levels. L1 and L2 are generally concerned with sensor fusion algorithms. Numerous sensor fusion studies exist in the literature including methods such as: Kalman filtering [141], sensor agreement [142], fault-tolerant abstract sensors [143], and fuzzy logic inference [144]. Most of these studies are focused on how to aggregate multi-sensor data from a technical perspective [145], i.e., communication protocols, saving data into databases, providing services, etc. L3 and L4 are concerned with the extraction of high-level knowledge (situation awareness, for example). When it comes to mHealth, not many studies are focused on the L3 and L4 data fusion [146]. Hristijan Gjoreski in his thesis focused on L3 and L4 fusion of information and knowledge from sensor data using classical ML algorithms [146]. In this thesis, we mostly focus on the fusion of ML models, including end-to-end deep learning, which is enabled by an additional meta-learning layer. Additionally, we propose an end-to-end DL architecture which performs L1/L2 type of fusion, by learning ML models directly from raw data which comes from multiple sensors.

A more thorough overview of the fusion and combining approaches used in the studies related to this thesis is presented in the related-work sections in each of the three studies presented in Chapter 5.

¹¹ Winner of the 2018 ACM Turing Award.

Chapter 3

Method for Mobile Health and Behavior Monitoring

This thesis proposes an ML method that extracts valuable human health and behavior information from wearable sensor data. The method uses end-to-end learning on sensor data as a standalone approach or in combination with classical ML to produce beyond state-of-the-art performance. Figure 3.1 shows the architecture of the proposed method. Each of the method's components is explained in the following subsections.

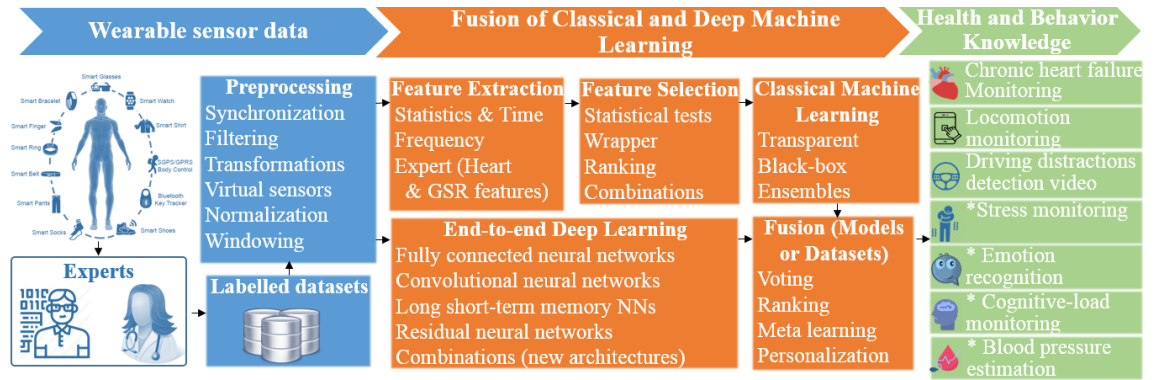


Figure 3.1: The architecture of the method for extracting human health and behavior information from wearable sensor data.

More detailed formal flowcharts and definitions are presented in the publications available in Chapter 5.

3.1 Sensor Data

The method uses as input any data collected using wearable sensors from human users. The type of sensors depends on the use-case. Regarding the sensors utilized in this thesis, in the studies on stress and cognitive load monitoring, physiological and acceleration data from a wrist-worn device was used. In the study on emotions, physiological data from wearable sensors was used. In the study on driving distractions, physiological and video-based sensors were used. In the study on locomotion recognition, smartphone sensors were used. In the study on blood pressure estimation, data from chest-worn ECG sensor was used. Finally, in the study on chronic heart failure (CHF) detection, digital stethoscope

was used to record heart sounds. Each of these studies had a different hardware setup, while the method is hardware-independent.

Regarding the data labels, in the study on CHF, the labels were provided by medical experts. In the rest of the studies, the labels were provided by the users themselves, e.g., by answering specialized questionnaires, or the labels were implicitly known from the experimental scenario. For example, in the study on driving distractions, the driving segments that contained distractions were predetermined. Choosing the right sensors, study design and labeling approach are crucial parts that have to be addressed as they significantly influence the outcomes of the specific studies.

Figure 3.2 presents example labeled data from the study on locomotion recognition. The graph at the top presents the labels (Subway, Train, Bus, Car, Bike, Run, Walk and Still). The rest of the graphs present sensor data from three smartphone sensors (accelerometer, gyroscope and gravity sensor). The vertical grey lines mark the different segments of the data, randomly selected from the Sussex-Huawei Locomotion dataset [28]. One segment represents 5 seconds data. The data of each sensor was sampled at a 100 Hz frequency.

3.2 Preprocessing

Once the labeled data is available, several preprocessing steps are required to minimize the noise in the data and to maximize the useful information. The following subsections describe the preprocessing steps required in an ML pipeline.

3.2.1 Data Synchronization

Usually, the data comes from a variety of sensors. In some cases, all the sensors are placed in a single device which performs the data synchronization, e.g., a wrist-worn device or a smartphone. However, in many cases, there are several devices that need to be synchronized. For example, in the study on driving distraction, there were 38 signals from five different devices: palm GSR from a palm device, heart rate and breathing rate from a chest device, eye tracking data, nasal EDA from a thermal camera, and facial expressions and emotions extracted from a normal camera. The data synchronization can be done using an external server's clock. In other cases, when the sensor devices do not have internal clocks, a joint event that can be recognized by all devices is used for synchronization. For example, to synchronize 15 wrist devices and video cameras, Ward et al. [147] used a 3-step process. First, all wrist devices were synchronized to a reference laptop. Second, the devices were enclosed in a padded box and the box was flipped sharply a few times to create distinctive spikes in the sensor data. Third, to synchronize the video with sensor data, the participants performed a warm-up dance which created distinctive signals in both the acceleration and the video.

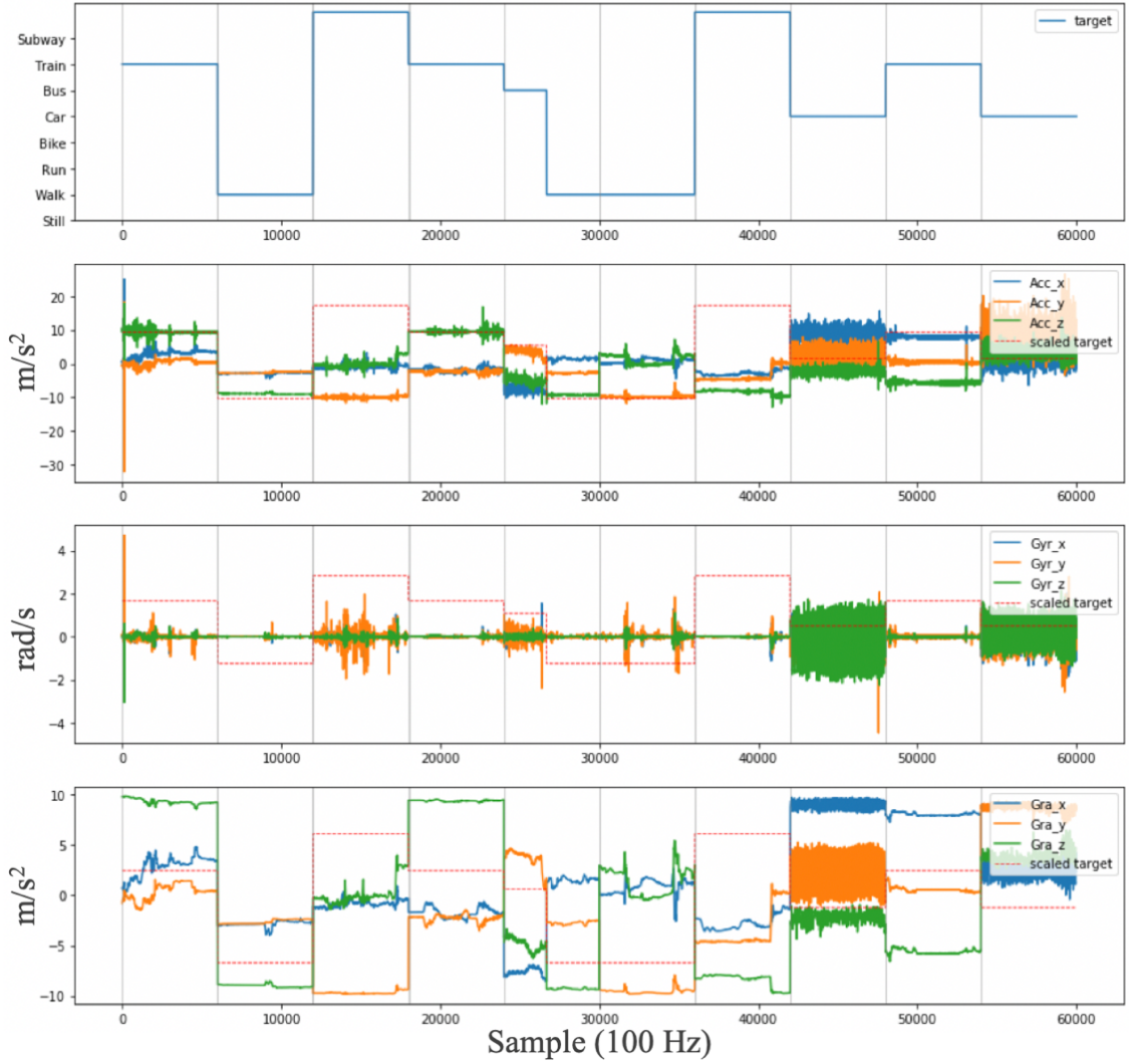


Figure 3.2: Labeled smartphone sensor data from the SHL dataset.

3.2.2 Filtering

The data from the wearable devices is quite often noisy. The usual source of the noise are movement artefacts and sensor misplacement. The filtering strategies include winsorization, detrending, moving average, low-pass, high-pass, band-pass, etc. **Winsorization** removes extreme values from the sensor data, e.g., values that are below the 1st percentile and above the 99th percentile.

Detrending is a filtering step that removes trends in the sensor data. Linear trends can be summarized by a linear model, and nonlinear trends can be summarized by a polynomial model. For the detrending procedure, curve-fitting methods can be used. One such method is the least-square fitting method which aims to minimize the squared sum of the residuals, i.e., differences between the fitted model and the sensor data. For example, if $\{x_i, t_i\}$ represents sensor data, where x_i is the i^{th} sample recorded at time t_i , a linear model $y = f(\mathbf{x}, \mathbf{p})$ can be fitted. In this model representation, $\mathbf{p}$ is a vector of parameters for the model. The residuals can be calculated using Eq. (3.1).

$$r_i(\mathbf{p}, x_i, t_i) = \text{abs}[x_i - f(x_i, \mathbf{p})] \quad (3.1)$$

An objective function that can be used by an optimization algorithm to obtain a least-squares fit using Eq. (3.2).

$$J(\mathbf{p}) = \sum_{i=0}^{N-1} r_i(\mathbf{p})^2 \quad (3.2)$$

where N is the number of sensor samples. Once $\mathbf{p}$ is determined, the fitted model f can be used to remove the trend in the data by subtracting the model's output from the original sensor data. Figure 3.3 presents a 1-minute ECG signal before (left) and after (right) detrending, and Figure 3.4 presents a 1-minute BVP signal before and after detrending. From the left graphs in the two figures, it can be seen that the trend in the two signals is not linear through the whole signal and is quite dynamic. To mitigate this problem, the signals were split into much shorter segments (e.g., few seconds), where the trend was linear, and then the detrending was applied. Our experiments showed that linear detrending on small segments produces better results than using polynomial detrending on the overall signal. Algorithm 3.1 presents the linear detrending algorithm used in our experiments.

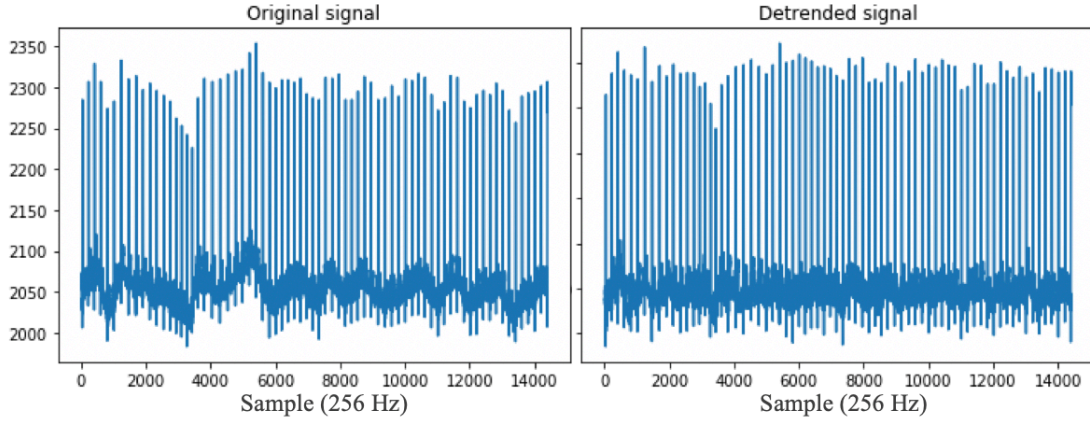


Figure 3.3: Filtering example. Detrended ECG signal.

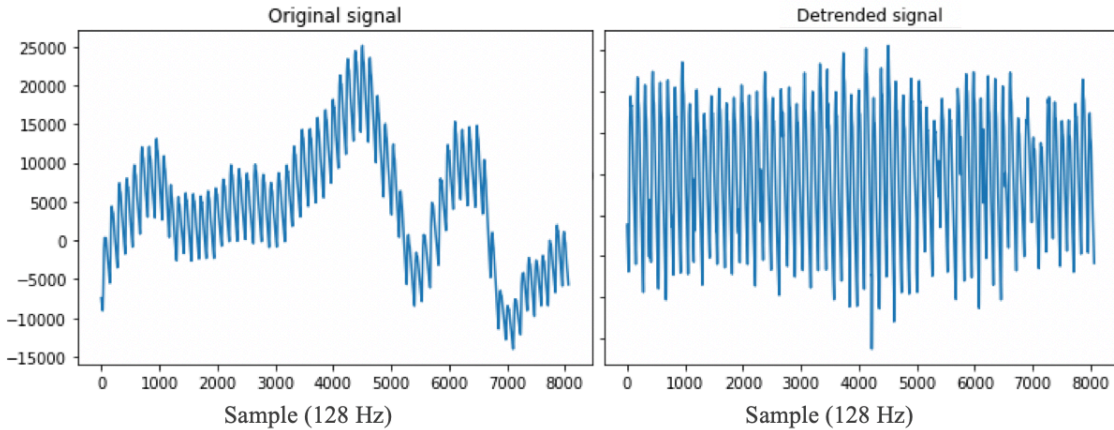


Figure 3.4: Filtering example. Detrended BVP signal.

Algorithm 3.1: Linear detrending using a sliding window.

```

Input: sensor_data, timestamps, sampling_frequency
Output: detrended data xf
win_size = 1 * sampling_frequency #use a window size of 1 second
win_start = 0
win_end = win_size
xf = copy(sensor_data)
while win_end < size(x) do
    x_values = sensor_data[win_start: win_end]
    x_time = timestamps[win_start: win_end]
    f = fit_linear_model(x_values, x_time)
    x_estimated = f.estimate(x_time)
    xf[start: end] = abs(x_values - x_estimated)
    win_start = win_start + win_size
    win_end = win_end + win_size
end
return xf

```

Moving average is a simple filter which replaces the values of each sample with the average value of that sample and its n closest neighbors (e.g., $n = 2$). Figure 3.5 depicts the effects of a moving average filter on data from a GSR sensor.

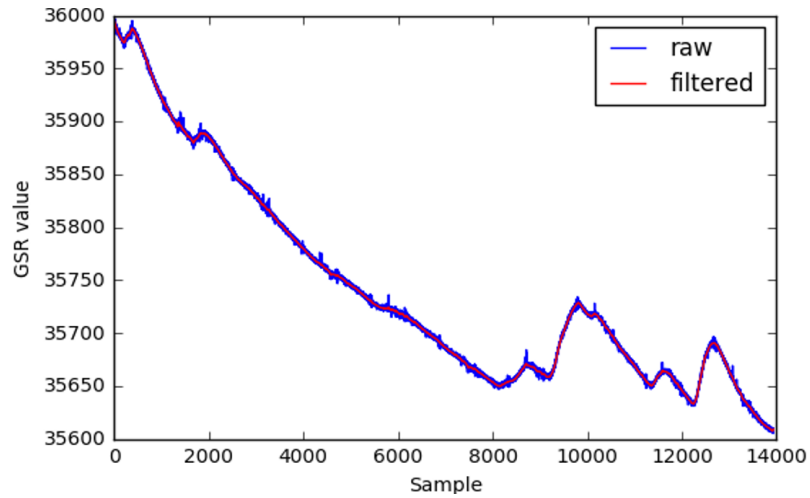


Figure 3.5: Example raw and filtered GSR signal. Used filter: moving average.

Low-pass, **high-pass** and **band-pass** filters are frequency-based filters that can be used to remove components from signals that appear at certain frequencies. As the name suggests, the low-pass filter keeps the components that have a frequency below a certain threshold. The high-pass filter keeps the components that have a frequency above a certain threshold. And the band-pass filter keeps the components that appear in a certain frequency band. These filters can be implemented as exponentially weighted moving average. Algorithm 3.2 represents one such implementation. Implementations based on Fast Fourier Transformation (FFT) and implementations based on Butterworth filter also exist.

Algorithm 3.2: Low-pass filter.

Input: sensor_data, smoothing_factor
Output: xf
 $\alpha = \text{smoothing_factor}$
 $\text{xf} = \text{copy}(\text{sensor_data})$
 $\text{xf}[0] = \alpha * \text{x}[0]$
for i **in** [1 to n] **do**
 $\text{xf}[i] = \alpha * \text{x}[i] + (1-\alpha) * \text{xf}[i-1]$
end
return xf

Figure 3.6 depicts the effects of a low-pass Butterworth filter applied on smartphone’s accelerometer data. A cut-off frequency of 25 Hz was used in this specific example. In the time domain, it can be seen that the filter smooths the signal, similarly to a moving average filter. In the frequency domain (Figure 3.7), it can be seen that before the filtering (left spectrogram), the signal contains components with a frequency of up to 50 Hz. After the low-pass filtering, the signal’s maximum frequency is 25 Hz. The exact filtering strategies depend on the sensors and the use-case. For example, in the CHF study, we applied a Butterworth filter [149] with a threshold of 1 kHz to the raw audio files, as most cardiovascular sounds are most likely to occur in the frequency range below 1 kHz [148].

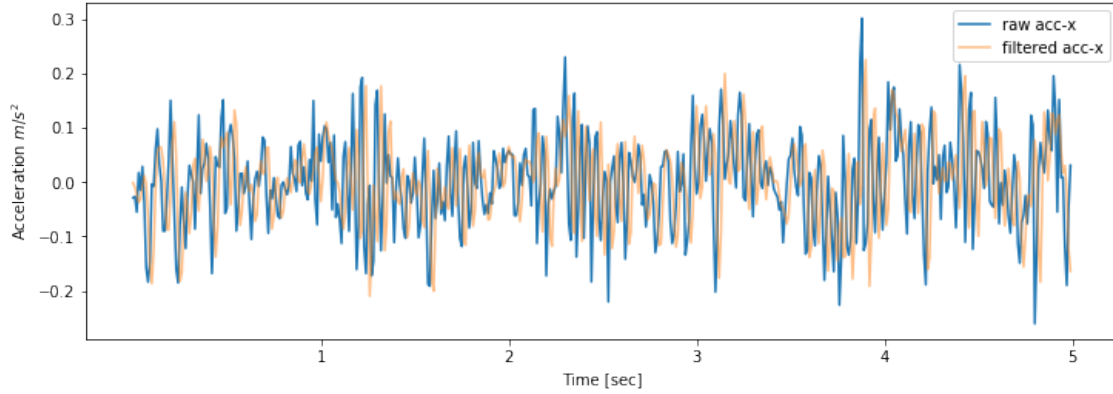


Figure 3.6: Raw and filtered acceleration signal. Used filter: low-pass.

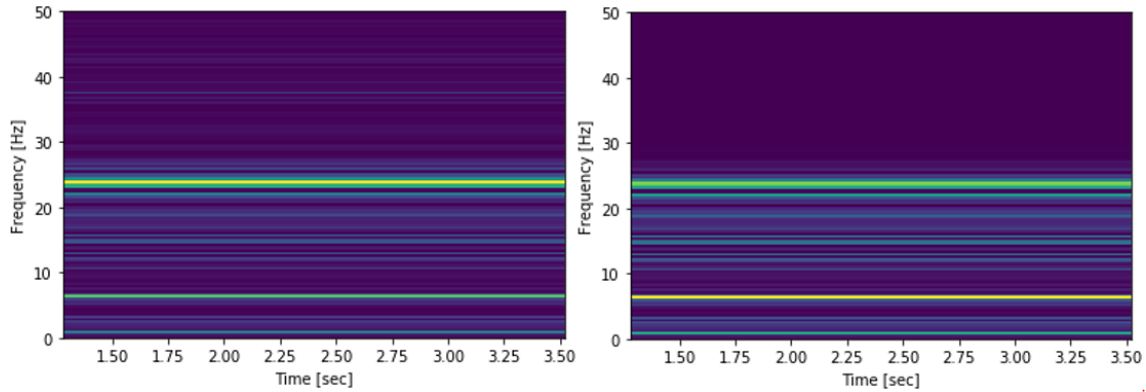


Figure 3.7: Raw (left) and filtered (right) acceleration signal. The color represents the amplitude of a particular frequency at a particular time. Used filter: low-pass.

3.2.3 Transformations and Virtual Sensors

The sensor data can be transformed into different domains (e.g., time domain and frequency domain), each of them specialized for extracting different types of information from the input data.

For example, by combining gyroscope and acceleration data from smartphone data, the acceleration data can be rotated. This produces location-independent acceleration data [27], which is useful for more robust activity recognition from smartphone sensors. Another simple transformation calculates the acceleration magnitude by combining the sensor from each axis (x-, y- and z-axis):

$$m_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \quad (3.3)$$

Definition 3.1 (Virtual sensors). Virtual sensors are sensors that use easily measured variables to estimate variables that are hard to measure due to technological limitations, large measurement delays, or high investment costs [183][184].

Definition 3.2 (R-R intervals). R-R intervals represent the time duration between two consecutive heartbeats.

Another typical transformation is the extraction of R-R intervals from sensors that measure heart activity (e.g., BVP and ECG sensors, see Figure 3.8). R-R intervals are much more informative compared to average heart rate data, as they allow better modeling of the heart dynamics. Algorithm 3.3 presents a simple implementation of a peak detection algorithm that is based on the signal's first derivative, a height (amplitude) threshold and a time threshold. The algorithm searches for the indices where the first derivative of the signal changes from positive to negative, which represents a candidate peak. Then, it checks whether the amplitude of the candidate peak is higher than the predefined height threshold. And finally, it checks whether the time constraint (time threshold) is also fulfilled. For example, for an average heart rate of 180 beats per minute, it is expected to discover three peaks each second (on average). Consequently, a time threshold close to 0.33 (330 milliseconds) would be a reasonable threshold for most studies that analyze human behavior under normal conditions.

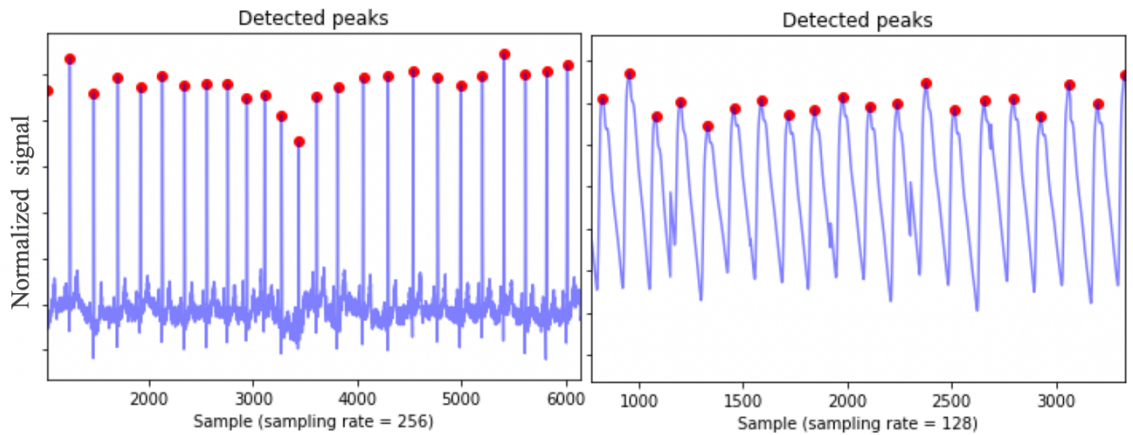


Figure 3.8: Example R-R intervals extracted from ECG (left) and BVP (right) signals.

Algorithm 3.3: Peak detection algorithm.

Input: sensor_data, sampling_frequency, height_threshold, time_threshold
Output: peaks
 $x = \text{sensor_data}$
 $\text{freq} = \text{sampling_frequency}$
 $\text{peaks} = []$
for i **in** $[0 \text{ to } \text{size}(x)]$ **do**
 $\text{diff} = x[i] - x[i - 1]$
 if $\text{prev_diff} > 0$ **and** $\text{diff} < 0$ **and** $x[i - 1] > \text{height_threshold}$:
 $\text{time_distance} = (i - \text{last_peak}) / \text{freq}$
 if $\text{time_distance} > \text{time_threshold}$:
 $\text{peaks.append}(i)$
 $\text{last_peak} = i$
 $\text{prev_diff} = \text{diff}$
end
return peaks

FFT is often used to transform sensor data into the frequency domain. According to the Fourier's analysis, any complex wave can be represented as a sum of simple sinusoidal waves with different frequencies. One way to represent a signal in the frequency domain is through its spectrum. For example, the following two figures show the representation of a signal in a time domain (Figure 3.9) and in a frequency domain (through its spectrum, Figure 3.10). The left graph in Figure 3.9 shows a complex signal in a time domain. The graph on the right shows how that complex signal can be represented as a sum of sinusoidal waves (blue color). Figure 3.10 shows how the signal spectrum is obtained from the sinusoidal waves. The spectrum is depicted on a graph (blue colors) with a vertical axis that represents the amplitude of the waves, and a horizontal axis that represents the frequency. The low frequency waves are represented at the beginning of the horizontal axis, and the frequency increases as we move to the right of the horizontal axis.

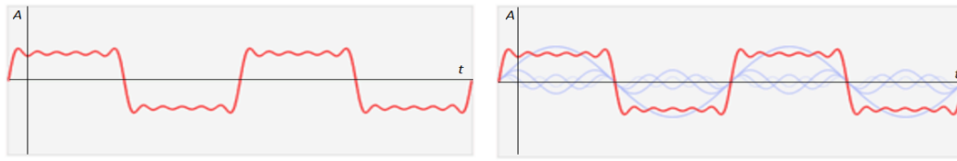


Figure 3.9: Decomposition of a complex signal into a sum of simple sinusoidal signals.¹²

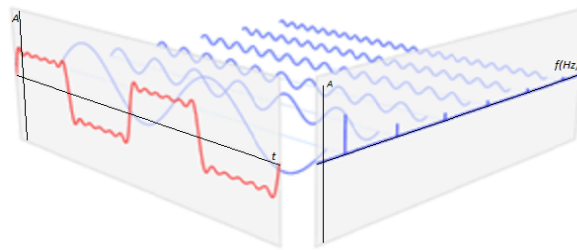


Figure 3.10: From time (red graph) to frequency (blue graph) domain.

¹² https://en.wikipedia.org/wiki/Frequency_domain

3.2.4 Normalization

Some people have a faster heart rate than other people, some sweat more than others, some walk faster than others, etc. This variability can harm any ML model, especially if there is a small dataset to train the ML model on. To minimize the inter-participant variability, different normalization (scaling) techniques can be employed. The normalization can be done either on the sensor data, or on features extracted from the data. In both cases, the normalization has similar effects, i.e., it scales the values of the variables, and in some cases, it also changes the distribution of the variables.

Figure 3.11 presents a scatterplot of two features (variables) calculated from two physiological sensors. The x-axis is Electrodermal Activity (EDA) which represents the average sweating intensity of the participants. The y-axis represents Root Mean Square of the Successive Differences (RMSSD), a feature often used as a Heart Rate Variability (HRV) measure. In addition to the scatterplot, the distribution of the two features is also presented using histograms (grey color). The EDA has a skewed distribution, which is expected because people sweat only when it is necessary. The skewness of the HRV distribution is not that explicit, but it is still different from a normal distribution. The scatter plot contains data from 12 participants (marked with different dot colors). From the colors, it can be seen that several clusters of participants can be easily identified. That type of clusters would be very useful if we were trying to identify the participants based on this data.

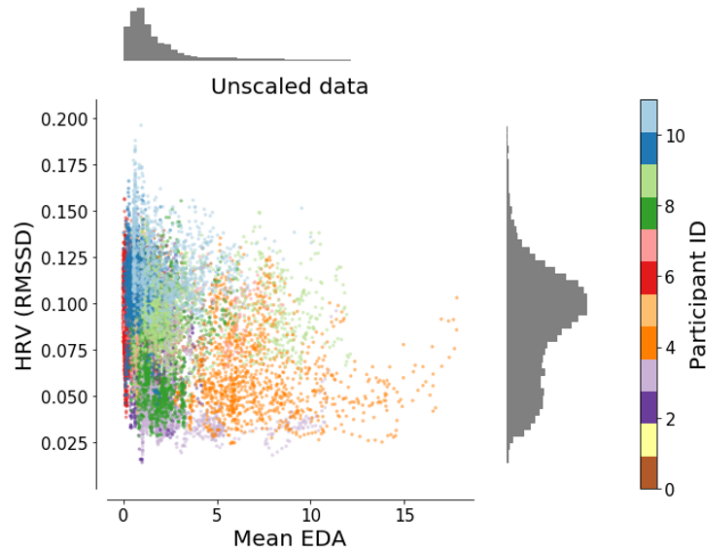


Figure 3.11: Data normalization (raw data).

However, in a typical HCI study, we would like this inter-participant variability to be minimized. The next several figures depict the effect that different normalization techniques have on the data. Eq. (3.4) presents min-max normalization:

$$x_i' = \frac{x_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \quad (3.4)$$

where, x_i is i^{th} data point of the feature $\mathbf{x}$. This normalization can be performed on the overall data (overall min-max scaling), or, it can be performed for each participant separately (personalized min-max scaling). Figure 3.12 presents personalized min-max scaling (left figure) vs. overall min-max scaling (right figure). It can be seen that for both

features, the normalization scaled the values between 0 and 1. However, the histograms show that the two approaches have a different effect on the data distribution. The overall min-max scaling does not change the feature distributions at all (compared to the unscaled data in Figure 3.11). On the other hand, the personalized min-max scaling changes the feature distributions, i.e., the distribution of the EDA is less skewed and the distribution of the HRV is closer to a normal distribution. However, the two distributions have a large variance. Additionally, from the colored scatterplots, it can be seen that after the personalized min-max scaling, the inter-participant variability is smaller and it is harder to identify clusters of participants.

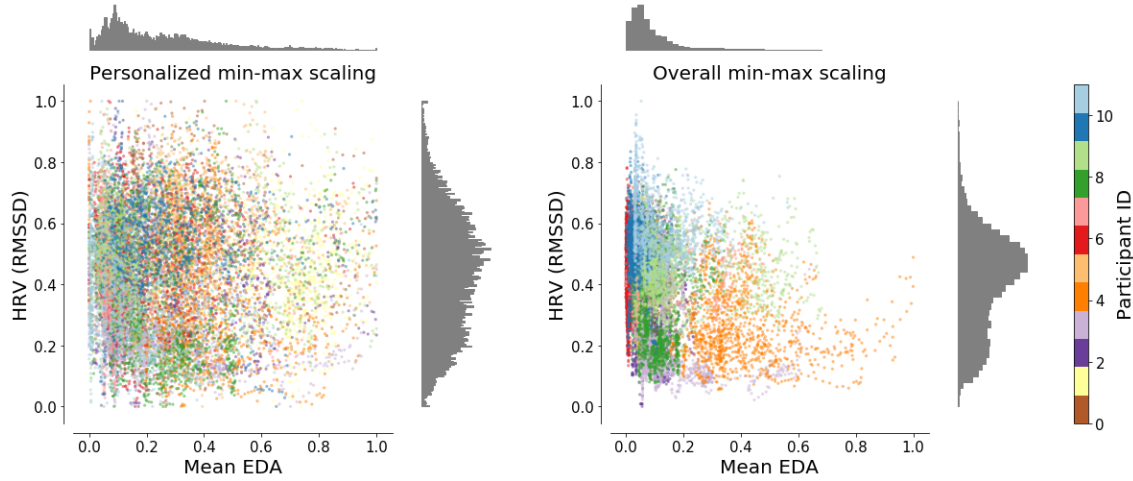


Figure 3.12: Personalized vs. overall min-max normalization.

Eq. (3.5) presents standard scaling (standardization), which normalizes the features by removing their mean and scales them to a unit standard deviation:

$$x_i' = \frac{x_i - \text{mean}(\mathbf{x})}{\text{stdDev}(\mathbf{x})} \quad (3.5)$$

Figure 3.13 presents personalized standardization (left figure) vs. overall standardization. Similarly to previous scaling, the overall standardization does not change the distribution at all, whereas the personalized standardization changes the distribution of the overall data. Compared to the personalized min-max scaling, the data here is even less skewed and has even lower variance.

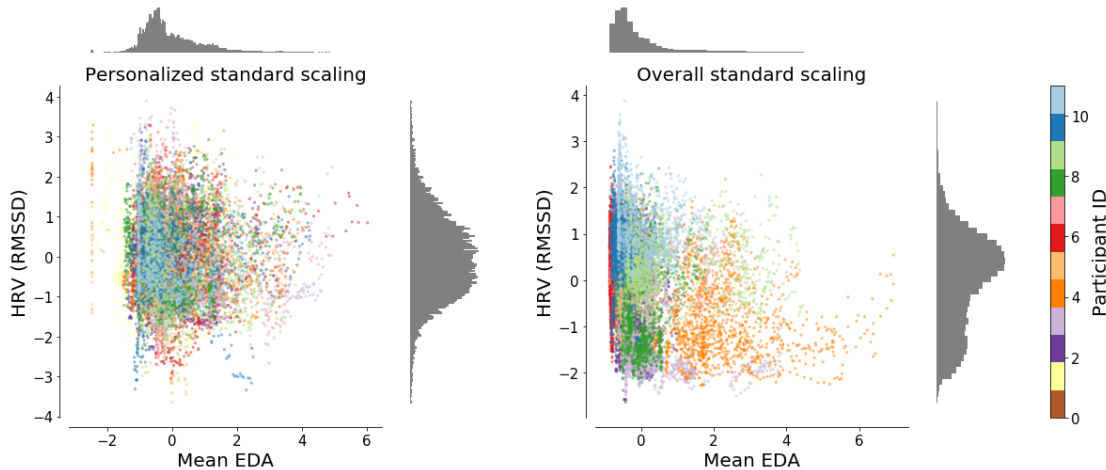


Figure 3.13: Personalized vs. overall standardization.

Outliers can often negatively influence the statistical measures used to scale the data (e.g., minimum, maximum, mean and standard deviation). For that reason, more robust measures can be used. For example, the median instead of the mean and the interquartile range instead of the standard deviation can be used. The effects of such scaling are presented in Figure 3.14. Interestingly, for the EDA, the personalized robust scaling boosted the values of some outliers further away from the general mean value. For the HRV feature, the distribution is similar as before, just on a smaller scale. The previous values were on a scale from -4 to 4 , and here the values are on a scale from -3 to 3 .

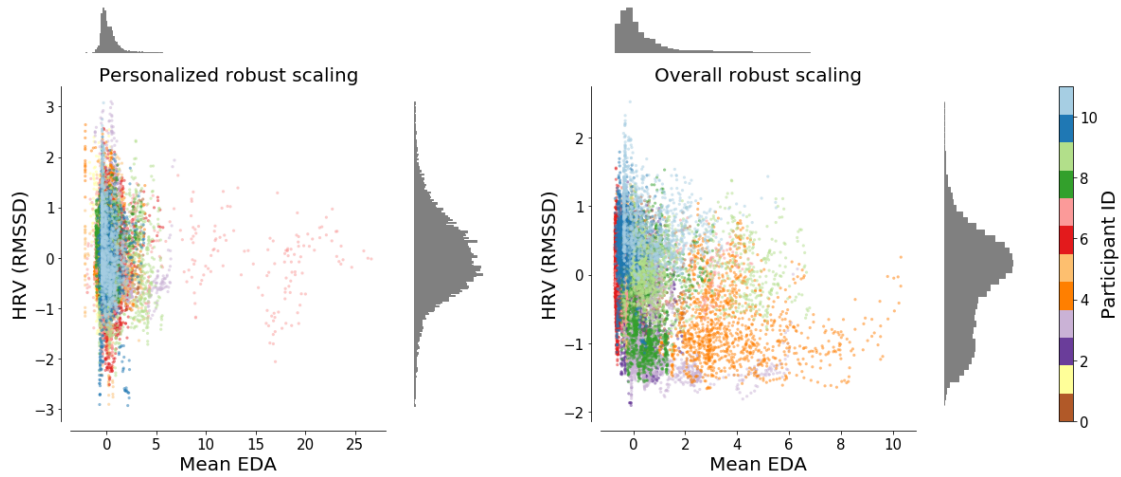


Figure 3.14: Personalized vs. overall robust normalization.

Another type of scaling typically used to remove outliers is winsorization – replacing extreme parts of a probability distribution at the low and high end with the most extreme remaining values. Figure 3.15 presents the effects of personalized vs. overall winsorization using a threshold of 5, i.e., the values below the 5th percentile are replaced with the 5th percentile, and the values above the 95th percentile are replaced with the 95th percentile. The first observation is that after the winsorization, the percentile-values appear as peaks in the data distribution. The personalized winsorization causes two peaks (low and high extremes) for each person, whereas the overall winsorization causes only two peaks overall. Another observation is that in the overall normalization, all orange data points for the EDA are skewed towards the same values (5). This may cause a loss of information for the “orange” participant.

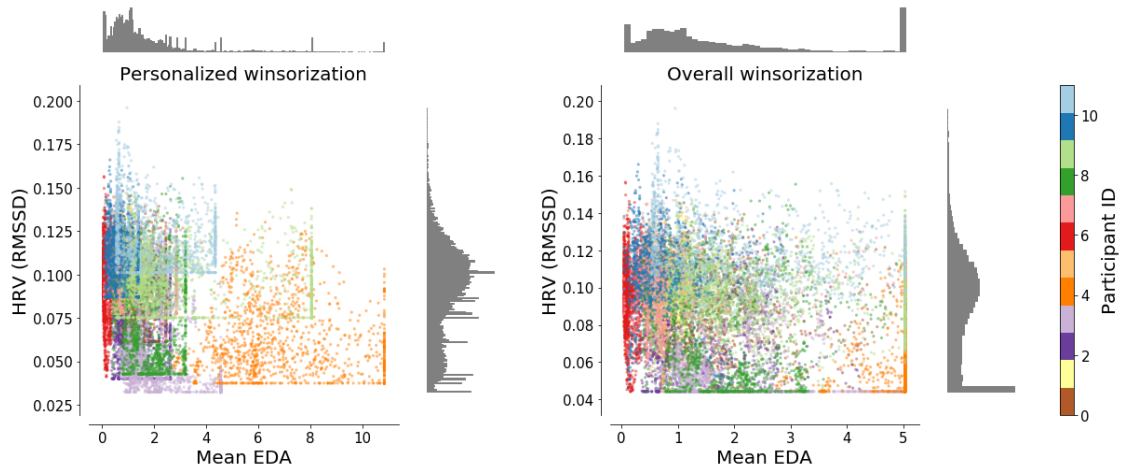


Figure 3.15: Personalized vs. overall robust winsorization.

There are also scaling techniques that ensure normal distribution of the transformed feature. This is done using the following steps. First, an estimate of the cumulative distribution function (CDF) of a feature is used to map the original values to a uniform distribution. The obtained values are then mapped to the desired output distribution (e.g., normal distribution) using the associated quantile function. CDF is defined as the probability that a random variable takes a value smaller than some specific value. The quantile function is the inverse of this function and returns the value to or below a given probability. Figure 3.16 depicts the effects of such transformation. It can be seen that both features have a normal distribution and in the personalized normalization, the inter-participant differences are minimized. The quantile transformation is non-linear, it tends to spread out the most frequent values and reduces the impact of outliers. It may distort linear correlations between variables measured at the same scale but renders variables measured at different scales more directly comparable. Additionally, some algorithms (e.g., linear and logistic regression) explicitly assume that the features have a normal distribution.

The most appropriate type of normalization depends on the data. Removing outliers from the data is always useful, but it should be performed carefully as sometimes the outliers may be the most informative data points. The safest way to select the right normalization approach is to treat it as a hyperparameter in the overall pipeline, which is tuned using an evaluation procedure.

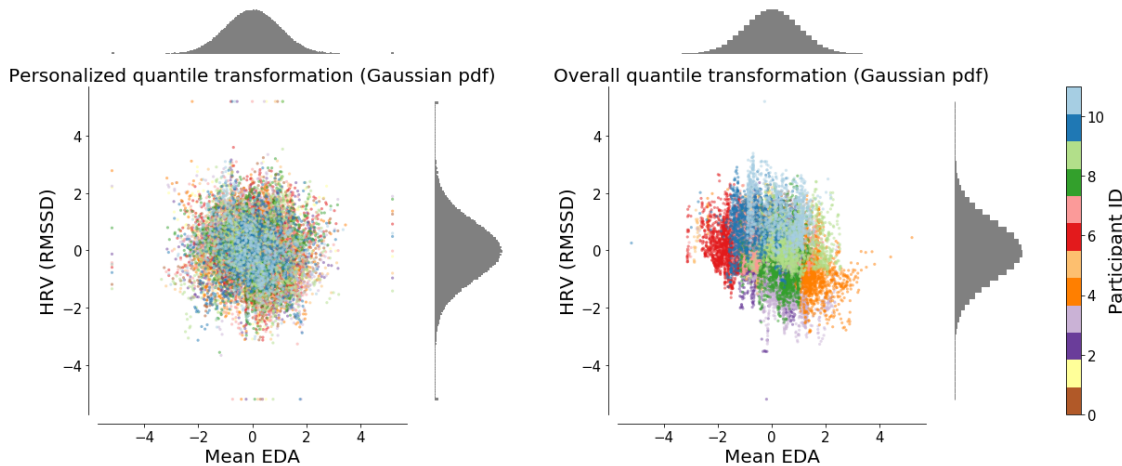


Figure 3.16: Personalized vs. overall quantile-based normalization (using the Gaussian probability density function).

3.2.5 Windowing

Wearable sensors typically provide a continuous stream of data. This data needs to be segmented into smaller windows in order to extract features, or to feed them into an end-to-end DL model. The size of the windows may significantly influence the performance of the ML models. Similarly to the normalization techniques, the window size should be treated as a hyperparameter in the overall pipeline. Another approach would be to combine several window sizes. For example, in the CHF study, we used a combination of 1-second windows and 30-second windows, which was the overall duration of one heart-sound recording. The 1-second windows contain at least one complete heartbeat (with common heart rates above 60 beats per minute). By extracting features both from the short windows and from the whole recordings, we aimed to capture both short-term and long-term information.

3.3 Feature Extraction

Informative features should be extracted which are used as input into classical ML algorithms. The feature extraction is an important step as it offers the possibility to encode expert knowledge into the system. In addition to the expert knowledge, for domains where expert knowledge is not well defined, all possible features can be extracted by “borrowing” expert knowledge from similar domains where the same sensors are utilized. In this thesis, we experimented with several types of features.

3.3.1 Statistical Features

Regardless of the sensor type and the problem domain, statistical features can always be extracted from the data. Example statistical features are: mean, standard deviation, skewness, kurtosis, 25th and 75th percentile, inter-quartile range, difference between the minimum and the maximum values and coefficient of variation, entropy, zero crossing rate, and autocorrelations (i.e., correlations of the signal with a delayed version of itself). Additionally, the same statistical functions can be applied on the signal’s first and second derivative. Thus, 30–40 statistical features can be easily extracted from each sensor stream. For sensors that have more than one stream (e.g., accelerometer), features that combine several sensor streams can also be used, e.g., Pearson’s correlation between the axes.

3.3.2 Frequency Features

These features are extracted from the spectral (frequency-based) representation of the sensor data. A typical approach is to calculate the power spectral density (PSD) of the signal, which is based on the FFT. Once the PSD is calculated, statistical features can be calculated from the PSD. For example, the lowest frequency in the PSD represents the signal’s fundamental frequency. The three largest magnitudes from the PSD represent the dominant frequencies in the signal. Both the magnitude values and the frequencies (in Hz) can be used as features. The binned distribution of the PSD can also be used as a feature, i.e., the distribution of the FFT magnitudes split into 10 equal-sized bins.

3.3.3 Heart-Related Features

The activation of the sympathetic nervous system triggered by emotional, cognitive, physical and other stressors yields more equidistant heartbeats. On the other hand, rest periods reverse this process, and the heartbeats become more irregular, as “A healthy heart is not a metronome” [167]. Heart Rate Variability Analysis (HRV) is commonly used to quantify the dynamics of the heartbeats (R-R intervals).

Figure 3.17 (left graph) presents an example of an R-R time series. The y-axis represents the duration of the R-R interval, and the x-axis the time (in seconds) in which the R-R interval has occurred. From the R-R signals, periodogram is calculated using the Lomb-Scargle algorithm [150]. The Lomb-Scargle algorithm allows efficient computation of a Fourier-like power spectrum estimator from unequally spaced data (as are the R-R intervals). Figure 3.17 (right graph) presents an example of the Lomb-Scargle periodogram. Example HRV features used in the literature are [151]: the mean heart rate (meanHR), the mean of the R-R intervals, the standard deviation of the R-R intervals (sdnn), the standard deviation of the differences between adjacent R-R intervals (sdsd), the square root of the mean of the squares of the successive differences between adjacent R-R intervals (rmssd), the percentage of the differences between adjacent R-R intervals that are greater than 20

ms, the percentage of the differences between adjacent R-R intervals that are greater than 50 ms, Poincaré plot indices (SD1 and SD2) [152], total spectral power of all R-R samples in power between 0.003 and 0.04 Hz (lf – low frequencies), between 0.15 and 0.4 Hz (hf – high frequencies), and the ratio of low to high frequency power.

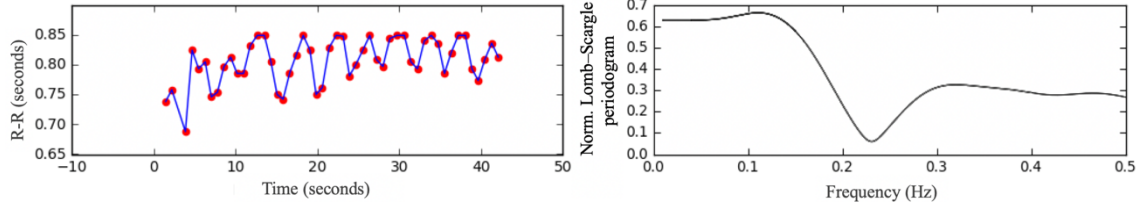


Figure 3.17: R-R signal in the time domain (left). Spectral representation (normalized Lomb-Scargle periodogram) of the same R-R signal (right).

Digital stethoscope can be used to record sounds that represent the activity of the heart. Figure 3.18 presents one such recording which is also known as phonocardiogram. From the figure, it can be seen that the R-R intervals can be detected, similarly to the ECG and BVP recordings. Consequently, all the features described in Subsection 3.3.3 can be extracted from the phonocardiogram as well.

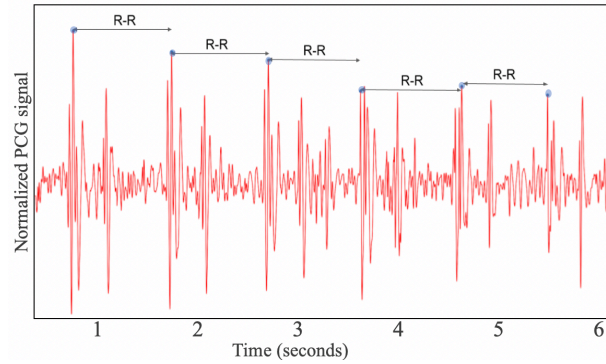


Figure 3.18: Example R-R intervals extracted from PCG signal (phonocardiogram).

In addition to the R-R intervals, general sound features can be extracted using sound-analysis techniques. The widely used features are as follows: mean, variance, skewness, kurtosis, power of the PCG signal, zero-crossing rate, Mel-frequency Cepstral Coefficients, Linear Predictive Coefficients, and Shannon entropy, statistics over the sound's power spectral density, jitter, shimmer, the Harmonics-to-Noise Ratio, etc. [108][109].

3.3.4 Galvanic Skin Response Features

GSR (or Electrodermal Activity – EDA) refers to electrical changes that arise when the skin receives specific signals from the brain. In particular, GSR is described by physiologists as produced purely by the sympathetic nervous system, thus GSR is argued to represent a sensitive measure of the activation of the sympathetic nervous system [154][155]. Consequently, GSR has been used in many studies to measure emotional activation, cognitive workload, physical exertion, etc. [11][45][153]. GSR can be measured using small electrodes attached to the skin. For example, the Empatica E4 wrist device measures GSR by flowing a minuscule amount of current between two dry electrodes, through which the device measures the conductivity on the wrist. The data unit is microsiemens (μS) and the frequency is 4 Hz.

Besides being a measure for average sweating levels, GSR is also characterized by Skin Conductance Responses (SCRs), which are caused by external stimuli (Figure 3.19). A typical SCR lasts 1–5 seconds, has a steep onset and an exponential decay or recovery, and reaches an amplitude of at least $0.1 \mu\text{S}$ [156][157]. Consequently, SCRs can be used as an additional measure of stimuli presence and stimuli intensity.

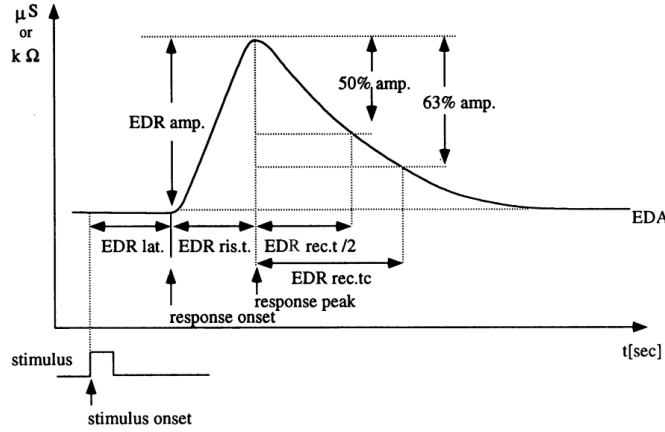


Figure 3.19: Skin Conductance Response.

Figure 3.20 represents a 5-minute segment of GSR data (blue line). The GSR signal can be decomposed into two components, a fast-acting component (SCRs) and a slow-acting component (tonic component). The SCRs can be detected using a peak detection algorithm (similar to Algorithm 3.3) with an additional extension, i.e., the peak onset (peak start, marked with green) and peak end (marked with yellow) need to be detected too. Since this has to be done automatically, some SCRs can be missed by the algorithm. For example, the purple dots represent rejected peaks, i.e., peaks that have been detected by the peak detection algorithm, but their corresponding start and end points were not detected.

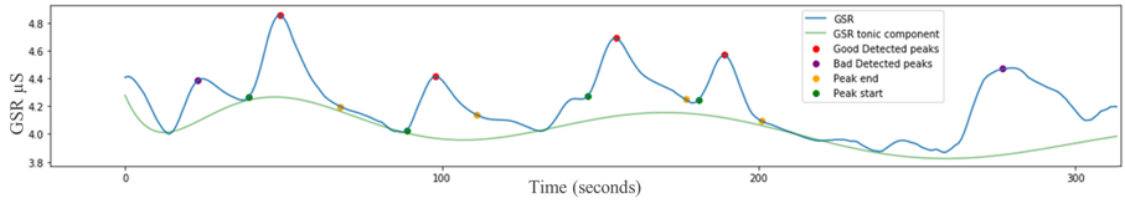


Figure 3.20: GSR signal with example Skin Conductance Responses (SCRs) detected.

The tonic component can be detected using polynomial curve fitting on the GSR data. Once the GSR data is decomposed, a variety of features can be extracted: the number of SCRs per second, the power of the SCRs, the sum of the signals' components that have a positive derivative, the ratio between the positive derivative and the negative derivative, the mean value of the derivative of the tonic, the mean value of the difference between the raw signal and the tonic component, the total spectral power of the signal in five frequency bands between 0 Hz and 0.6 Hz with a 0.1 Hz span, the amplitude increase of the largest SCR (from the SCR start time to the SCR peak), the amplitude decrease of the largest SCR peak, the largest SCR increase time, the largest SCR decrease time, the ratio of the increase time and the decrease time of the largest SCR peak, the largest SCR duration, the

largest SCR peak increase and decrease slope, the average amplitude increase and decrease of all SCR peaks, and the average amplitude change of all SCRs.

3.4 Feature Selection

In general, feature selection methods can be divided into:

- Wrapper methods – use an ML model to score different feature subsets. Each new subset is used to train a model, which is then tested on an evaluation set;
- Ranking methods (also known as filter methods) – use a scoring function to rank features' quality for the specific domain (e.g., correlation between features and data labels);
- A combination of wrapper and combination methods;
- Embedded methods – perform feature selection as part of the model-learning process (e.g., the LASSO ML algorithm penalizes uninformative features by setting their weights to zero).

Wrapper methods require an ML algorithm and an evaluation metric (e.g., accuracy, ROC [168], Bhattacharyya distance [169], etc.). Wrapper methods provide better results compared to the ranking methods, especially if the ML algorithm is pre-defined. However, if pre-defining an ML algorithm is not possible, the selected feature set is biased toward the ML algorithm used by the wrapper method. Additionally, wrapper methods induce a heavy computation burden as they re-train the ML model for each evaluation. On the other hand, the ranking methods are computationally efficient and require only a scoring metric (e.g., mutual information [170], information entropy [171], Fisher score [172], Wilcoxon signed-rank test [173], etc.) to evaluate the quality of the features. Most of the ranking methods are limited to the scoring of only single features, and as a consequence, some informative combinations may be missed. Figure 3.21 presents two features (dimensions) that are not informative when analyzed separately because they have the same distribution for the two classes (blue and red). However, when analyzed jointly, the two features provide good separation between the two classes.

In our study on monitoring stress, a new feature selection method was proposed by combining ranking and wrapper methods. The method aims to minimize the number of evaluations required by the wrapper method by prioritizing top features ranked by information gain, and by removing low-ranked features which are correlated with the top-ranked features. Correlation between the features is expected because several features are extracted from one data source (e.g., several features are extracted from the BVP signal, several from the R-R signal, etc.). The method includes several main steps: rank the features by the information gain, calculate the correlation coefficients between features, evaluate different subsets of features (selected based on information gain rankings and correlation coefficients) using the leave-one-subject-out (LOSO) technique, and finally provide the best-performing feature set. The pseudocode is presented in Algorithm 3.4.

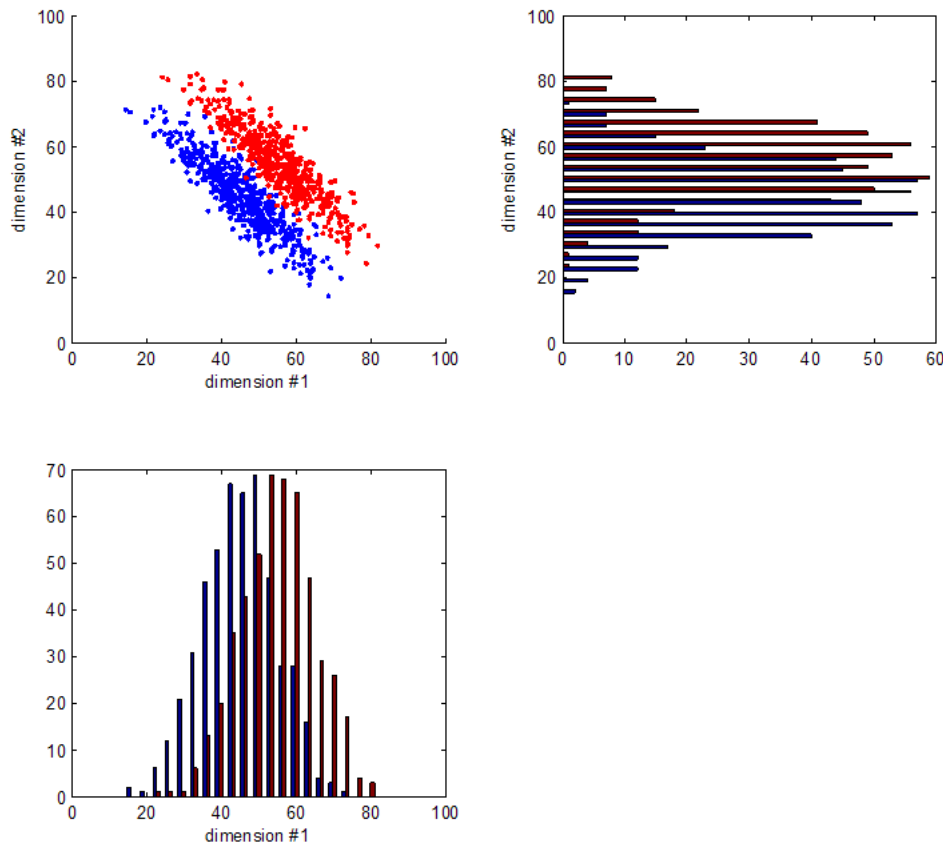


Figure 3.21: The combination of two non-informative features can be informative.¹³

3.5 Classical Machine Learning

The extracted features can be fed into a variety of classical ML algorithms including algorithms that produce comprehensible classifiers (e.g., Decision Trees), black-box classifiers (e.g., SVM) and ensembles (e.g., Extreme Gradient Boosting). In most of our studies, we used the following algorithms:

1. **Majority classifier** – always predicts the majority label. This is a baseline model.
2. **J48** – an algorithm for building a decision tree (DT) [174].
3. **Naïve Bayes classifier** – an algorithm for building simple probabilistic classifiers based on Bayes' theorem with strong independence assumptions between the features [175].
4. **KNN** – an algorithm which provides a prediction based on k-nearest instances (from the training set) to the test instance [176].
5. **SVM** – an algorithm for building a classifier where the classification function is a hyperplane in the feature space [177].
6. **Bagging** – an ensemble algorithm that learns base models on subsets of train data with the purpose of reducing variance and avoiding overfitting [178].

¹³ https://en.wikipedia.org/wiki/Data_fusion

Algorithm 3.4: Feature selection algorithm.

```

Input: instances, ML_algorithm
Result: best_features
current_features = getFeatures(Instances)
best_features = current_features
best_score = evaluate(instances, ML_algorithm, current_features)
rankings = calculateInfoGain(instances)
sorted_rankings = sort(rankings)           # rank features by increasing info-gain
size = size(sorted_rankings) * 0.9         # check the lowest 90%
while counter < size do
    test_feature = sorted_rankings[counter]
    remove_flag = false
    correlation_coefficients = pearsonsCorrelation(instances, test_feature)
    for each correlation_coefficient in correlation_coefficients:
        if correlation_coefficient > 0.8      # check if there is a strong correlation
            removeFlag = true
            current_features = removeFeature(current_features, test_feature)
            break
    if removeFlag
        current_score = evaluate(instances, ML_algorithm, current_features)
        if current_score >= best_score
            best_score = current_score
            best_features = current_features
    counter++
end
return best_features

```

7. **Boosting** – an ensemble algorithm (AdaBoost and Extreme Gradient Boosting – XGB) that learns models on subsets of the train data and “boosts” the weights of misrecognized instances allowing for the models in the ensemble to focus on the misclassified instances [179].
8. **Random Forest** – an ensemble method that learns base models by subsetting the feature set [180].

Most of the algorithms require hyperparameter tuning to achieve the best possible performance. Algorithm 3.5 presents a pseudocode for the hyperparameter tuning performed in our studies. The algorithm is based on random hyperparameter search. A more advanced approach would be, instead of the blind hyperparameter search, to use informed search based on the history of previously evaluated hyperparameters. Such informed search can be implemented using Bayesian-optimization techniques [181].

 Algorithm 3.5: Algorithm for hyperparameter tuning.

```

Input: ML_algorithm, hyperparameters, instances, max_iterations
Output: optimized_parameters
max_score = 0      # the bigger this score, the better the model (e.g., accuracy)
while i < max_iter do:
    tmp_param = generateRandHyperparameters (ML_algorithm, hyperparameters)
    tmp_score = evaluateKFold(ML_algorithm, tmp_param, instances)
    if tmp_score > max_score:
        max_score = tmp_score
        optimized_parameters = tmp_param
end
return optimized_parameters
  
```

3.6 End-to-End Deep Learning for Sensor Data

Processed sensor data can be used to build end-to-end DL models, i.e., models that do not require feature extraction and can learn directly from sensor data with the potential to discover new useful patterns in the data, previously unknown to experts. DL models include existing DL architectures (e.g., Convolutional Neural Networks [159], Long Short-Term Memory (LSTM) Neural Networks [139], their combination – ConvLSTMs [160], Residual Networks – ResNet [161], etc.), and our novel DL architecture Spectro-Temporal Residual Network (STResNet). STResNet is a novel DL architecture for end-to-end learning specialized for multimodal sensor data. More specifically, STResNet can learn from several sensors simultaneously, each of them having a different sampling frequency, and it learns both in the time domain and in the frequency domain. The mathematical background of the artificial neural networks is presented in Chapter 5, Sections 5.1 and 5.2.

To present some intuition about how DL models learn directly from raw data, Figure 3.22 visualizes patterns learned by a simple end-to-end CNN model with two CNN layers and one fully connected layer. Figure 3.22 shows two examples (one left and one right) from an end-to-end CNN model for arousal recognition from the GSR signal. The top-left graph shows the normalized GSR input for the network. It can be seen that the maximum value of the signal is 0.6 which means it is near the participant’s average sweating rate. Also, the signal is dropping continuously, which may be a sign that the participant is relaxing (low arousal). The middle-left graph shows the output of a convolutional layer for the given input. This layer contains 32 different filters. Each filter produces different representation of the input signal. Each of the 32 representations are marked with a different color. From the 32 different representations, it can be seen that the CNN layers are emphasizing the peaks of the signal and that some of the representations are delayed, i.e., the signal’s peaks are shifted in time. This time shift may produce temporal independence, i.e., in certain scenarios, the exact timing of the peaks is not important, but rather their height and their duration. These findings are in line with many physiological and affective-computing studies which analyze Skin Conductance Responses (SCRs), one of the main features for affect recognition [70][71][23]. The bottom left graph represents the input to the fully connected layer. It contains 2912 input values, which correspond to the 32 different representations (CNN filters) multiplied by 91 – which is the length of the input signal after the convolutions. The bottom arrow represents the final output of the model, which is an estimation for the input to belong to a certain class. In this case, high

vs low arousal. For the sample on the left, the prediction is “low arousal” with an estimated probability of 54% and in this case, it was a correct estimation.

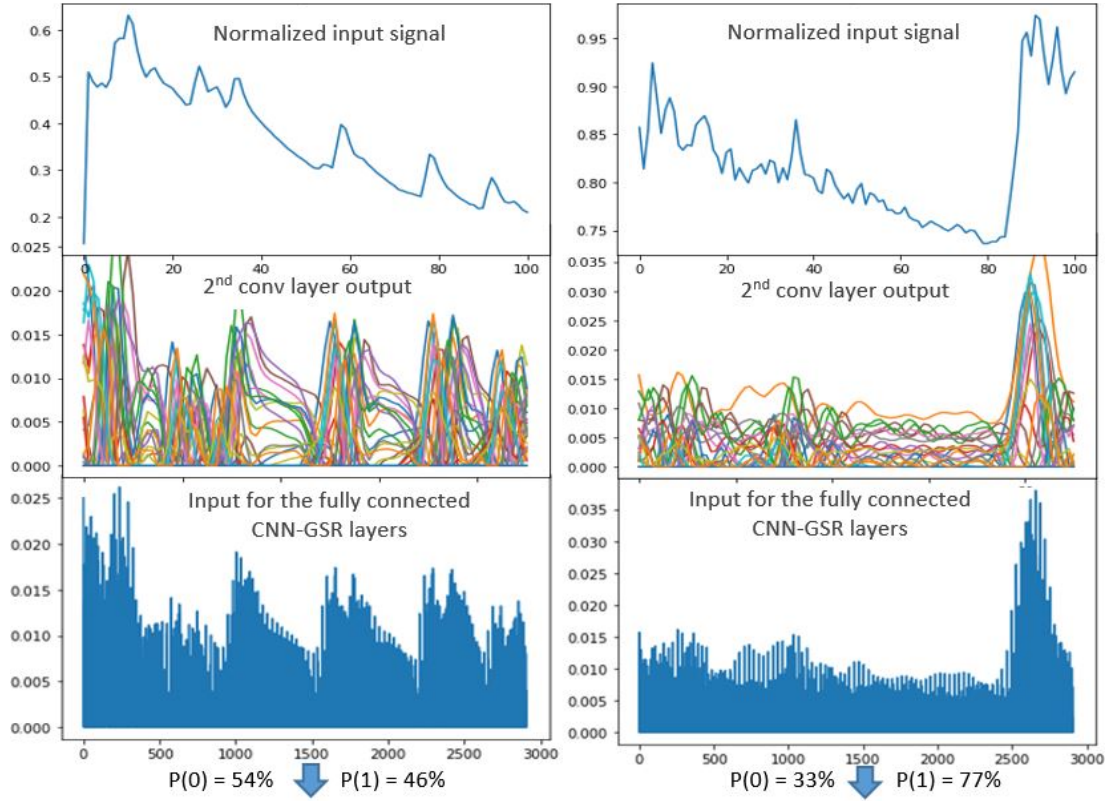


Figure 3.22: Two examples from an end-to-end CNN model for arousal recognition from the GSR signal. The top graphs represent the normalized GSR input.

The right side of Figure 3.22 presents another example of the same model, however, in this case for a “high arousal”. From the input (top-right), it can be seen that the values of the signal are over 80%, which is a high sweating rate close to the participant’s maximum. In addition to that, there is an abrupt positive change towards the end of the signal, which may indicate affective reaction of the person to a certain stimulus. The middle-right graph shows that the CNN layers emphasized the high peak of the signal. Finally, the prediction of the model (bottom-right arrow) is that the signal is a “high arousal” signal with an estimated probability of 77%, which was a correct prediction. The middle graphs represent the output of CNN layers. The bottom graphs represent the input for a fully connected layer.

3.7 Fusion of Models and Datasets

Following the principle of multiple knowledge [162], classical ML models and DL models can be fused to provide the final output.

Definition 3.3 (Information fusion). Information fusion is the process of integrating multiple information sources to produce more consistent, accurate, and useful information than that provided by any individual source [163].

Depending on the use-case, in some cases, only the highest-ranked model is used to minimize complexity. In other cases, classical meta-learners and voting ensembles are used to maximize accuracy. In third cases, meta-learners that can account for temporal dependencies in the data are used. For example, we used LSTMs and Hidden Markov Models as meta-learners for activity recognition to avoid prediction errors of the type: walking, walking, walking, car, walking, walking, etc. In this example, it is obvious that the prediction “car” should be replaced with a prediction “walking”. In some cases, personalized meta-learners were used. For example, in the study on blood pressure estimation from ECG signals, a personalized meta-model was used to adapt the prediction of the general models to each participant.

Besides the model-fusion approaches, we also explored with dataset fusion approaches using a naïve approach, a multi-task learning and a transfer learning approach. In the naïve approach, a joint model is learned simultaneously from several datasets. In the multi-task learning approach, a hybrid model is learned in parallel from several datasets. Typically, the hybrid model has components that learn shared representations for the several datasets, and components that learn dataset-specific representations. We experimented with multi-task learning for cognitive-load monitoring by learning joint models over data collected in two separate studies [37][38]. In the transfer learning approaches, typically a model is pre-trained on one dataset (a source dataset), and the training is later finalized on another target dataset. We experimented with transfer learning for human activity recognition by transferring models between four different activity-recognition datasets [39].

Definition 3.4 (Multi-task learning). Multi-task learning is a sub-field of machine learning that improves learning for one task by using the information contained in the training signals of other related tasks [164].

Definition 3.5 (Transfer learning). Transfer learning is a sub-field of machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. Transfer learning techniques provide mechanisms for transferring instances, feature representations, model parameters and/or relational-knowledge [165][166].

Chapter 4

Early Work

This chapter presents a summary of the early work performed for this thesis, i.e., work that was performed in the beginning of the thesis process (monitoring stress using context [23], non-invasive blood pressure estimation from ECG signals [33], and arousal recognition from physiological signals [40]), and work that is still in progress (cognitive-load inference from physiological signals) [36][37]. The common characteristic of all these studies is that they used a variety of physiological signals and a combination of classical machine learning methods. Each study is also published in a corresponding journal paper. In the following sections, each study is presented separately, and the final section presents a general summary and discussion.

4.1 Monitoring Stress Using Context

In this study, a method for stress detection that can accurately, continuously, and unobtrusively monitor psychological stress in real life was developed. First, the problem of stress detection was explored using machine learning and signal processing techniques in laboratory conditions, and then the extracted laboratory knowledge was applied to real-life data. For that purpose, a novel context-based stress-detection method was developed (see Figure 4.1) using data from a wrist device that provides the acceleration (marked as Acc. Data in the figure), and BVP, EDA, HR, R-R and skin temperature data (marked as Bio. Data in the figure).

Two datasets were collected in this study: one in a laboratory environment to analyze the problem under controlled conditions, and one in real life without any restrictions to the participants' normal life. The summary of the datasets is presented in Table 4.1.

Table 4.1: Experimental data for monitoring stress.

	Lab Data	Real-life Data
# Participants	21	5
Age Mean	28±4 years	28±4.3 years
No Stress	840 minutes	1216 hours
Low Stress	356 minutes	70 hours
High Stress	368 minutes	41 hours

The method consisted of three machine-learning components: a laboratory stress detector that is trained on laboratory data and detects short-term stress every 2 min; an activity recognizer that continuously recognizes the user's activity and thus provides

context information; and a context-based stress detector that uses the outputs of the laboratory stress detector, activity recognizer and other contexts (e.g., day of the week and time of day), in order to provide the final decision on 20-min intervals.

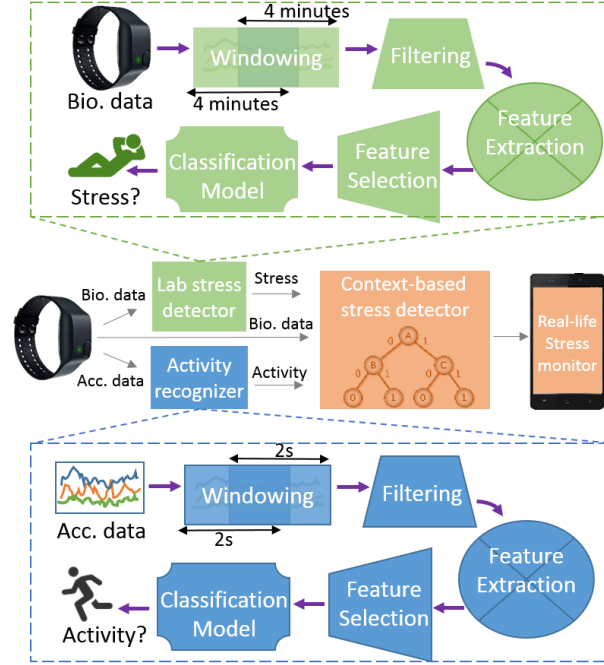


Figure 4.1: Context-based stress detection.

Figure 4.2 presents the evaluation results for the laboratory stress detector on the lab data using the leave-one-subject-out (LOSO) evaluation technique. Each bar represents a different feature set used for the experiments. The subsets were selected per sensor. One feature set was generated for the PPG sensor, which includes BVP, HR, and IBI data, and one additional feature set was generated using the feature-selection algorithm described in Section 3.4. The best performance was achieved when the algorithm used the combination of all sensors. When sensor-specific features were used, the IBI and HR sensor data performed slightly better than the other sensor-specific feature sets.

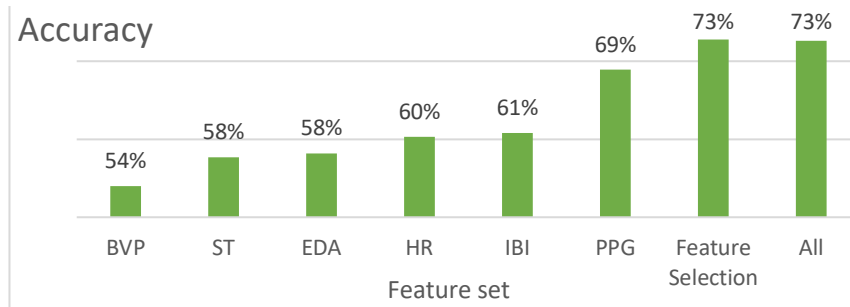


Figure 4.2: Evaluation results for the laboratory stress detector. LOSO evaluation.

Next, the overall context-based method was tested using LOSO evaluation technique on the real-life dataset. The results are presented in Table 4.2. It can be seen that the context-based classifier performed significantly better than a no-context classifier. The no-context classifier was the laboratory classifier applied to the real-life data. For example, the context-based classifier achieves a mean F1-score of 90% (mean value of 0.99 and 0.81 in percentage points), and the no-context classifier achieves a mean F1-score of 47%.

Table 4.2: Confusion matrices and evaluation metrics (Precision, Recall and F1 score) for No-Context vs. With-Context stress detection.

	No-Context		With-Context	
	No Stress	Stress	No Stress	Stress
No Stress	3308	1630	4932	6
Stress	34	125	47	112
Precision	0.99	0.07	0.99	0.95
Recall	0.67	0.79	1.00	0.70
F1 score per class	0.80	0.13	0.99	0.81
Mean F1 score (%)		47		90

4.2 Non-Invasive Blood Pressure Estimation from ECG

In the blood pressure study, a method was developed for BP estimation using only ECG signals. The raw ECG data was filtered, segmented, and features were extracted based on complexity analysis. Then, a machine-learning method was applied, by stacking a classification module for predicting BP classes (e.g., normal, prehypertension, and hypertension), and a regression module for estimating systolic BP (SBP), diastolic BP (DBP), and mean arterial pressure (MAP). The MAP is calculated directly from the SBP and DBP values using Eq. (4.1). In addition, the method included a probability distribution-based calibration to adapt the models to particular users.

$$MAP = \frac{SBP + 2 * DBP}{3} \quad (4.1)$$

The experimental dataset contained close to 26 hours of labelled data from 51 participants between 16 and 74 years old. For each participant, cuff-based blood pressure measurement and a corresponding ECG recording were available.

The proposed method is depicted in Figure 4.3. The raw ECG signals are segmented into 30-second segments, each accompanied with SBP, DBP and MAP values. Those values pass through a preprocessing step in which a band-pass filter is applied to preserve only valid ECG information. Following that, the signals are forwarded to the module for feature extraction. Next, the feature vectors are input into the classification module, which implements a stacking ML approach. The output of the classification module, in combination with the extracted features, is input into a regression module, which outputs the BP estimate. The last module is a calibration module, which allows for probability distributions-based calibration.

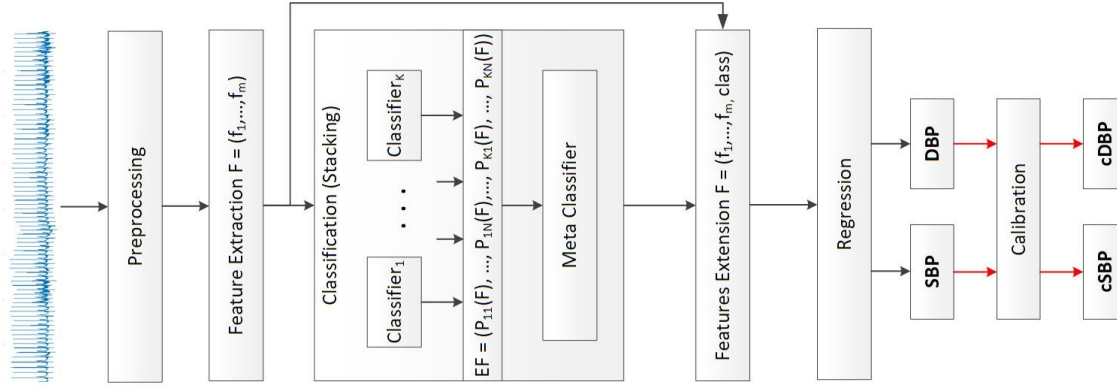


Figure 4.3: Method for non-invasive blood pressure estimation from ECG.

By following the train-validation-test principle, the data were randomly split into three different non-overlapping datasets: 60% of the subjects were included in the training, 10% in the validation and 30% in the testing set. The experimental results are presented in Table 4.3. The method achieved a mean absolute error (MAE) of 8.64 mmHg for SBP, 18.20 mmHg for DBP, and 13.52 mmHg for the MAP prediction. When the models were calibrated, the MAE decreased to 7.72 mmHg for SBP, 9.45 mmHg for DBP, and 8.13 mmHg for MAP. The results indicate that when the probability distribution-based calibration was used, the proposed method could achieve results close to those of a certified medical device for BP estimation.

Table 4.3: Mean absolute error for cuff-less blood pressure estimation.

	No Calibration	Calibration
Systolic Blood Pressure (SBP)	8.64	7.72
Diastolic Blood Pressure (DBP)	18.2	9.45
Mean Arterial Pressure (MAP)	13.52	8.13

4.3 Emotion Recognition from Physiological Signals

In most of the studies on emotion recognition, only one dataset is analyzed. In this study, we performed experiments with six different datasets (domains), we analyzed which ML algorithms in combination with which modality, i.e., an independent source of data (either R-R intervals or GSR in this study), yields best performance across all six datasets for arousal recognition, and we proposed a novel meta-learning method for learning from all six domains jointly.

The data belonged to six publicly available datasets for emotion recognition: Ascertain [189], DEAP [190], Driving workload dataset [191], Cognitive load dataset [23], Mahnob [192], and Amigos [193]. Overall, we analyzed nearly 150 hours of arousal-labelled data that belonged to 191 subjects (see Table 4.4). In some datasets, the subjects had one long trial (e.g., 20-minute driving session) under varying conditions, and in other datasets the subjects had several short trials (e.g., they watched 30 different 1-minute videos). The datasets were processed and translated into a common spectro-temporal space of R-R intervals and Galvanic Skin Response (GSR) data, from which features were extracted and fed into ML algorithms.

Table 4.4: Experimental datasets for the study on emotion recognition.

Dataset	Subjects	Females	Average Age	# Trials	Duration [h]
Ascertain	58	21	31	36	46.4
DEAP	32	16	26.9	40	21.3
Driving	10	3	35.6	1	5.0
Cognitive	21	0	28	2	28.0
Mahnob	30	17	26	40	26.7
Amigos	40	13	28	16	15.3
Overall	191	70	29.25	135	142.7

First, we evaluated which of the two modalities (R-R and GSR) is more informative for arousal recognition. Table 4.5 presents the accuracy achieved by dataset-specific Random Forest models for the two modalities analyzed in the study. The evaluation was performed using trial-specific 10-fold cross-validation, i.e., the data segments that belong to one trial (e.g., one affective stimulus) can either belong only to the training set or only to the test set; thus there was no overlapping between the training and test data. We also tried LOSO evaluation, but the performance for all ML models was low. From the results, it can be seen that there is no clear answer to the question “Which modality yields the best performance across all six different datasets?”. For three datasets (Ascertain, DEAP and Amigos), the models based on the R-R modality outperformed the models based on the GSR modality. For one dataset (Driving), the models achieved equal accuracy, and for the two datasets Cognitive and Mahnob, the models based on the GSR outperformed the models based on the R-R intervals. In general, the combination of the two modalities achieved the best (Amigos, Mahnob, Cognitive, Driving) or close to best performance (DEAP and Ascertain). In the following experiments, only the combination of the two modalities was used.

Table 4.5: Accuracy for dataset-specific models for different modalities (R-R intervals, Galvanic Skin Response and the combination of the two).

Dataset	R-R	GSR	R-R + GSR
Ascertain	65	64	64
DEAP	55	50	54
Driving	78	78	78
Cognitive	73	76	76
Mahnob	58	61	61
Amigos	53	47	56
Average	64	53	65

Next, we evaluated the novel meta-learning scheme depicted in Figure 4.4, which was designed to train a meta-learner that utilizes the knowledge from all six datasets. In the example scenario, seven ML algorithms were used. Thus, there are 42 base models (6 datasets $\times$ 7 ML algorithms). The outputs of the base models, which are probabilities for the class “high arousal”, are used as input to a meta-learner. The meta-learner can be any supervised ML algorithm.

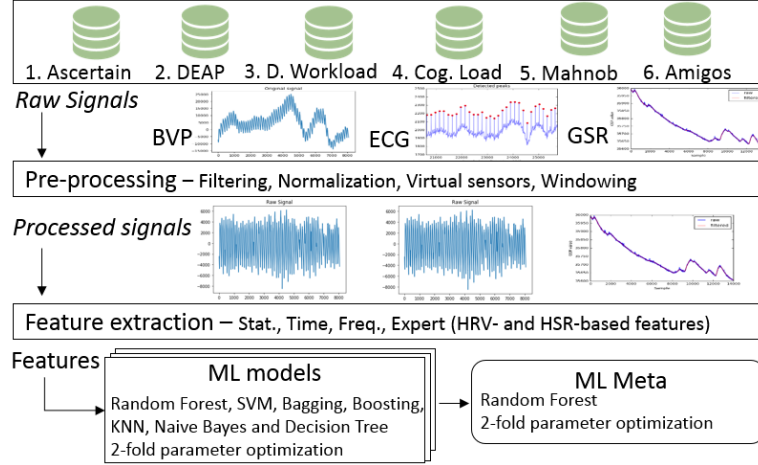


Figure 4.4: Meta-learning for emotion recognition from physiological signals.

We compared the meta-learning approach to “flat” ML algorithms trained on all merged datasets. The evaluation is performed using the same trial-specific 10-fold cross-validation. The results are presented in Table 4.6. It can be seen that the meta-learning improved upon the results of the “flat” models. All of the “flat” algorithms achieved an accuracy below or equal to 60%. The meta-learning approach improved the results by achieving an accuracy of 63%.

Table 4.6: Accuracy (%) for emotion recognition from physiological sensors.

RF	SVM	GB	AdaB	KNN	NB	DT	Meta	Majority
58	60	59	58	56	60	59	63	51

4.4 Cognitive-Load Inference from Physiological Signals

In this study, we examined the potential for inferring one's cognitive load using merely cheap wearable sensing devices. We subjected 23 volunteers to varying cognitive load using six different tasks. As metrics for the cognitive load, we used one subjective measure, the NASA Task Load Index (NASA-TLX), and one objective measure, designed task difficulty. Additionally, the participants' physiological response was recorded using a wrist device measuring heart rate, R-R intervals, GSR, and skin temperature.

The physiological data were preprocessed and filtered, features were extracted, and ML models for cognitive load prediction were constructed. The ML models were task-specific, i.e., a separate model was built for each of the six different tasks. Table 4.7 presents the results obtained using LOSO evaluation and Random Forest models for each modality separately, and their combination. It can be seen that the models that use the combination of all modalities (R-R, GSR and skin temperature) perform the best.

Table 4.7: Accuracy (%) for different modalities (R-R intervals, Galvanic Skin Response, Skin Temperature and the combination of the three) on the Cog. Load dataset.

R-R	GSR	Skin Temperature	Combination
65	55	55	68

In a similar study, Timotej Knez for his bachelor thesis [194] collected another dataset for cognitive load. In the study, 23 participants played the game Snake¹⁴ on smartphones. The game’s task difficulty was varied (easy, medium, and hard) by changing the game speed. The participants’ physiological response was recorded using the same wrist device (measuring heart rate, R-R intervals, GSR and skin temperature).

We analyzed the two datasets, Cog. Load and Snake, in a joint study. ML models for predicting task difficulty were developed using a variety of ML algorithms. Each model was developed using a processing pipeline that includes preprocessing, segmentation, feature extraction, normalization, feature selection, and model learning. In addition to the classical ML algorithms, a multi-task learning (MTL) neural network was compared to a single-task learning neural network (STL).

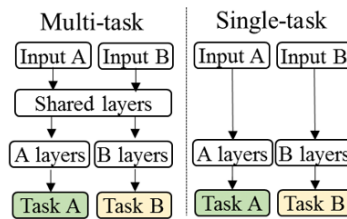


Figure 4.5: Multi-task learning vs. single-task learning.

The results are presented in Table 4.8. In general, the ML models performed better on the Snake dataset compared to the Cog. Load dataset, because the Cog. Load models were person-independent, and the Snake models were person-dependent. The highest accuracy of 82.3% on the Snake dataset was achieved by the XGB algorithm in combination with feature selection and feature standardization. The highest accuracy of 68.2% on the Cog. Load dataset was achieved by the Bagging algorithm in combination with feature selection and feature standardization. Another observation is that the ensemble models (e.g., Bagging, and XGB) performed better compared to the single-model algorithms, which is probably because the ensemble models are more robust to noise.

Regarding the proposed MTL approach, it is interesting to note that both MTL and STL performed similarly for the dataset that contained more instances (the Cog. Load dataset). However, for the smaller dataset, the MTL approach outperformed the STL approach, which may indicate that combining similar datasets using MTL is useful when the target dataset is small.

Table 4.8: Cognitive-load inference from physiological signals. Accuracy for best-performing algorithm (bold) vs. single-task-learning (STL) vs. multi-task learning (MTL) for two experimental datasets, Snake and Cog. Load.

Dataset	Model	Accuracy (%)
Cog. Load	Bagging	68
	STL	63
	MTL	65
Snake	XGB	82
	STL	66
	MTL	70

¹⁴ [https://en.wikipedia.org/wiki/Snake_\(video_game_genre\)](https://en.wikipedia.org/wiki/Snake_(video_game_genre))

4.5 Summary and Discussion

This chapter presented four studies in which the proposed general method was applied, to improve the performance of the baseline models for extracting human physical, physiological, and psychological information from the wearable sensor data. **Table 4.9 presents a summary of the general method’s components (depicted in Figure 3.1)**, which were used in the presented studies. The row “Experts” presents which type of experts collaborated in the specific study (ML experts, medical experts, psychologists and cognitive scientists). The word “All” in the row on preprocessing and in the row on feature extraction indicates that all steps from the specific components were used. The word “Many” in the row on ML indicates that a variety of ML algorithms were evaluated and compared. DL was used in the three studies presented in Chapter 5. The row on fusion indicates how different models and datasets were fused. In three of the four studies, meta-learning was used to enable the fusion of several ML models. In one study, multi-task learning was used to build a joint model for two separate datasets. In all of the studies, the fusion of sensors, datasets, or models improved the results over the baselines.

Table 4.10 presents a summary of the experimental results. The experimental results are not directly comparable across the different studies because of the many differences in the experimental setups. The differences include the following: different dataset sizes, different sensors, different preprocessing steps, different classification tasks, different evaluation procedure, etc. However, by using the study-specific baseline methods as a comparison proxy, common conclusions can be observed.

Table 4.9: Method summary. Presenting which components of the proposed general method (Figure 3.1) were used in the early-work studies.

Method’s component	Study			
	Stress	Blood pressure	Emotions	Cog. Load
Sensors	Smartphone + physiological	Physiological	Physiological	Physiological
Dataset	Ours	Ours + public dataset	Public datasets	Ours
Experts	ML + psychology	ML + medical	ML	ML + cognitive sci.
Preprocessing	All	All	All	All
Feature Extraction	All	All	All	All
Feature Selection	Wrapper + ranking	Ranking	Ranking	Ranking
ML	Many (classification)	Many (classification + regression)	Many (classification)	Many
DL	/	/	/	/
Fusion	Meta (lab. stress detector + AR classifier)	Meta (BP prediction + calibration)	Meta (stacking)	Multi-task learning

The column “Combining modalities” summarizes the results for different modalities. It can be seen that in all studies, the combination of several modalities outperformed the best single modality, which experimentally **confirms the first hypothesis for the three studies in which more than one modality was available**, i.e., combining data from a variety of wearable sensors improves the performance for monitoring human physical, physiological and psychological states using machine learning.

The column “Combining methods or datasets” summarizes the results for meta-learning vs. no meta-learning. In the stress study, the context-based method outperformed the no-context method by utilizing a meta-learner which combined predictions from several classifiers. In the blood pressure study, the meta-learner trained on person-specific data improved the results from the no-calibration methods. In the emotion recognition study, the meta-learner achieved higher accuracy compared to the baseline by utilizing knowledge from six datasets. Similarly, in the cognitive-load study, the MTL neural network outperformed a single-task neural network by simultaneously learning from two datasets, which experimentally **confirms the second hypothesis for the four studies**, i.e., our method for combining machine-learning models enables additional meta-learning, which improves the performance for monitoring human physical, physiological and psychological states. However, it should be noted that in the cognitive-load study, the best-performing ensemble methods outperformed the proposed MTL, which may indicate that an ensemble MTL approach may improve the results even further.

Table 4.10: Results summary. The numbers in the table represent the evaluation score used in each study (Accuracy, F1-score, or Mean Absolute Error).

Study (metric)	Method	Combining modalities	Combining methods or datasets
Stress (Acc./F1.)	Baseline	69 (Accuracy, Figure 4.2)	47 (mean F1-score, lab. stress detector, Table 4.2)
	Comb.	73 (Accuracy, Figure 4.2)	90 (mean F1-score, lab. stress detector + AR classifier, Table 4.2)
Blood pressure (MAE)	Baseline	/	13.52 (MAE for Mean Arterial Pressure, no calibration, Table 4.3)
	Comb.	/	8.13 (MAE for Mean Arterial Pressure, with calibration, Table 4.3)
Emotions (Acc.)	Baseline	64 (Table 4.5)	60 (dataset-specific classifiers, Table 4.6)
	Comb.	65 (Table 4.5)	63 (a stack of dataset-specific classifiers, Table 4.6)
Cog. Load (Acc.)	Baseline	65 (Cog. Load, Table 4.7)	66 (Snake) and 63 (Cog. Load), (STL, Table 4.8.)
	Comb.	68 (Cog. Load, Table 4.7)	70 (Snake) and 65 (Cog. Load) (MTL, Table 4.8.)

Chapter 5

The Fusion of Classical and Deep Machine Learning

This chapter presents three studies on combining classical and deep machine learning applied to sensor data for mobile health and behavior monitoring. Each study is presented in a separate section via its corresponding journal publication, containing an introduction to the domain problem, related-work analysis, data description, method description, experimental results, discussion and conclusion. Additionally, the proposed STResNet is introduced and utilized in all of the three studies. The method used in each study is a variation of the general method presented in Chapter 3.

Section 5.1 presents a study on the detection of chronic heart failure (CHF) via heart sounds. This study was performed in collaboration with the University Medical Center in Ljubljana. The models developed in that study were evaluated on data from real CHF patients in Slovenia. This system is currently being tested on data from new patients, and the performance of the system achieves high accuracies, similar to the performance reported in the paper.

Section 5.2 describes a study on monitoring driving distractions via physiological and visual sensors. This study was performed in collaboration with the Fraunhofer Institute of Integrated Circuits (IIS), Erlangen, Germany. It aimed at utilizing our knowledge in processing wearable sensor data and IIS's knowledge of image processing.

Section 5.3 presents a study on recognizing the means of transport (locomotion). This study thoroughly analyzes JSI-Deep, the method that combines classical ML and DL models for AR from smartphone sensors. Additionally, it analyzes the energy-efficiency of the ML methods using methods developed by Janko in his PhD thesis [195].

Finally, a Section 5.4 summarizes and discusses the findings of the three studies at a more general level.

5.1 Chronic Heart Failure Monitoring

Received December 3, 2019, accepted January 3, 2020, date of publication January 23, 2020, date of current version January 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968900

Machine Learning and End-to-End Deep Learning for the Detection of Chronic Heart Failure From Heart Sounds

MARTIN GJOESKI^{1,2}, ANTON GRADIŠEK¹, BORUT BUDNA¹, MATJAŽ GAMS^{1,2}, AND GREGOR POGLAJEN^{3,4}

¹Jozef Stefan Institute, SI-1000 Ljubljana, Slovenia

²Jozef Stefan Postgraduate School, SI-1000 Ljubljana, Slovenia

³Advanced Heart Failure and Transplantation Program, Department of Cardiology, UMC Ljubljana, SI-1000 Ljubljana, Slovenia

⁴Medical Faculty, University of Ljubljana, SI-1000 Ljubljana, Slovenia

Corresponding author: Martin Gjoreski (martin.gjoreski@ijs.si)

This work was supported in part by the Slovenian Research Agency under Grant U2-AG-16/0672 - 0287 and by the EU project PlatformUptake.eu.

ABSTRACT Chronic heart failure (CHF) affects over 26 million of people worldwide, and its incidence is increasing by 2% annually. Despite the significant burden that CHF poses and despite the ubiquity of sensors in our lives, methods for automatically detecting CHF are surprisingly scarce, even in the research community. We present a method for CHF detection based on heart sounds. The method combines classic Machine-Learning (ML) and end-to-end Deep Learning (DL). The classic ML learns from expert features, and the DL learns from a spectro-temporal representation of the signal. The method was evaluated on recordings from 947 subjects from six publicly available datasets and one CHF dataset that was collected for this study. Using the same evaluation method as a recent PhysoNet challenge, the proposed method achieved a score of 89.3, which is 9.1 higher than the challenge's baseline method. The method's aggregated accuracy is 92.9% (error of 7.1%); while the experimental results are not directly comparable, this error rate is relatively close to the percentage of recordings labeled as "unknown" by experts (9.7%). Finally, we identified 15 expert features that are useful for building ML models to differentiate between CHF phases (i.e., in the decompensated phase during hospitalization and in the recompensated phase) with an accuracy of 93.2%. The proposed method shows promising results both for the distinction of recordings between healthy subjects and patients and for the detection of different CHF phases. This may lead to the easier identification of new CHF patients and the development of home-based CHF monitors for avoiding hospitalizations.

INDEX TERMS Chronic heart failure, deep learning, heart sounds, machine learning, PCG.

I. INTRODUCTION

Chronic heart failure (CHF) is a chronic, progressive condition underscored by the heart's inability to supply enough perfusion to target tissues and organs at the physiological filling pressures to meet their metabolic demands [1]. CHF has reached epidemic proportions in the population, as its incidence is increasing by 2% annually. In the developed world, CHF affects 1-2% of the total population and 10% of people older than 65 years. Currently, the diagnosis and treatment of CHF uses approximately 2% of the annual healthcare budget. In absolute terms, the USA spent approximately 35 billion

USD to treat CHF in 2018 alone, and the costs are expected to double in the next 10 years [2]. Despite the progress in medical- and device-based treatment approaches in the last decades, the overall prognosis of CHF is still dismal, as 5-year survival rate of this population is only approximately 50%. In the typical clinical course of CHF, we observe alternating episodes of compensated phases, when the patient feels well and does not display symptoms and signs of fluid overload, and decompensated phases, when symptoms and signs of systemic fluid overload (such as breathlessness, orthopnea, peripheral edema, liver congestion, pulmonary edema) can easily be observed. During the latter episodes, patients often require hospital admission to receive treatment with intravenous medications (diuretics, inotropes) to achieve a

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Liu.

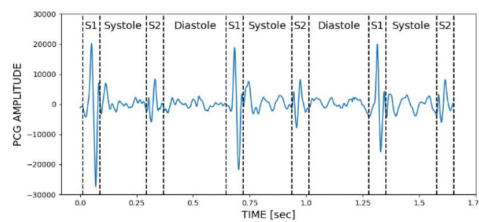


FIGURE 1. Example PCG recording with the PCG regions of interests (S1, S2, Systole and Diastole) marked.

successful negative fluid balance and return to the compensation state. Early detection of HF worsening would allow a treating physician to adjust the patient's medical management on an outpatient basis in a timely manner and thus avoid the need for a hospital admission. Currently, an experienced physician can detect the worsening of HF by examining the patient and by characteristic changes in the patient's heart failure biomarkers, which are determined from the patient's blood. Unfortunately, clinical worsening of a CHF patient likely means that we are already dealing with a fully developed CHF episode that will most likely require a hospital admission. Additionally, in some patients, characteristic changes in heart sounds can accompany heart failure worsening and can be heard using phonocardiography. An example of a phonocardiogram (PCG) recording of a healthy subject is presented in Fig. 1. In healthy subjects, 2 heart sounds are typically heard (called S1 and S2). S1 is caused by the closure of the mitral valve and ventricular wall in the early systole, S2 is caused by the closure of the aortic and pulmonary valves at the beginning of the diastole. Here, the interval between S1 and S2 is called systole, i.e., the contraction phase of the cardiac cycle, and the interval between S2 and S1 is called diastole, i.e., the relaxation phase of the cardiac cycle. Additional heart sounds (such as S3 and S4) can be heard in certain cardiac conditions and are never regarded as normal. In the case of CHF (in the course of decompensation), we can often hear a third sound (S3) that typically appears 0.1-0.2 s after the second sound, i.e., S2.

Recently, it has been demonstrated that some physiological parameters, such as the occurrence of additional heart sounds or increased blood pressure in the pulmonary circulation, already start to appear several weeks before the CHF patient develops a clinically evident decompensation episode. This is also an important therapeutic window where outpatient-based treatment interventions can reverse CHF deterioration and return the patient to the compensated state without the need for a hospital admission.

In recent years, many studies have proposed Machine-Learning (ML) approaches for the automatic detection of different heart conditions using PCG signals recorded with a digital stethoscope [1]. Nevertheless, methods that explicitly focus on CHF detection are quite scarce. The typical ML pipeline for the detection of different heart conditions is as follows: segmentation of the signals by detecting the "typical" heart sounds (i.e., S1 and S2), denoising of the

signals, extracting individual frequency-domain and time-domain features, and learning a feature-based ML model (e.g., using ML algorithms, such as Random Forest or Support Vector Machine - SVM) that is capable of classifying healthy vs. unhealthy sounds. Most of the features currently used are based on medical and audio/signal analysis knowledge.

However, a PCG recording that sounds unhealthy to one expert may sound healthy to another one; therefore, doctors never diagnose a CHF patient using only heart sounds, but rather use a holistic view of the patient instead (i.e., extensive medical history, blood pressure, laboratory tests, etc.). This uncertainty is one reason why 9.7% of the recordings in the recent PhysioNet cardiology challenge [3] were actually labeled as "unknown" by experts, while the rest of the recordings were labeled as healthy or unhealthy.

The recent advancements in Deep Learning (DL) suggest that end-to-end learning (i.e., ML models that learn directly from the raw data and no features are needed) can outperform the classic, feature-based ML. For example, DL has achieved breakthrough performance in tasks such as pattern recognition problems [4], image processing [5], [6], natural language processing [7], [8], speech and audio processing [9], [10], and sensor data processing [11], [12]. For CHF detection, a successful combination of classic ML and end-to-end DL can outperform each single approach [13]. The classic ML approach learns from a large body of expert-defined features, and the DL approach learns both from a time-domain (the raw PCG signal) representation of the signal and a temporal-domain representation (the spectrogram) of the signal. This approach was successful in our previous study of human activity recognition from smartphone sensor data [14].

In addition to distinguishing the CHF patients and healthy individuals, we focus on detecting the CHF state (compensated vs. decompensated) based on the analysis of heart sound recordings. Our work builds upon the initial studies, where we demonstrated that it is possible to distinguish between healthy individuals and patients in a decompensated CHF episode using a stack of machine-learning classifiers and expert features, showing promising results on a limited dataset [15]. We expand upon this approach using a considerably larger patient dataset, including six additional PhysioNet datasets, and an improved ML method that uses end-to-end DL. Furthermore, we investigate the differences in the heart sounds during the transition between the decompensated and re-compensated states of CHF, with the aim of developing personalized monitoring models. Early detection of the worsening of CHF has the potential to reduce hospitalizations due to the worsening of the condition, which both improves the quality of life of patients and decreases the financial and logistic burden on the patient and the health system.

II. RELATED WORK

A typical ML pipeline includes segmentation of the signals, denoising of the signals, extracting individual frequency-domain and time-domain features, and

learning ML capable of classifying healthy vs. unhealthy sounds [1], [3]. Regarding the segmentation process, Schmidt *et al.* [16] developed an algorithm (later improved by Springer *et al.* [17]) that segments the signals into the following four stages: S1, S2, systole, and diastole. The algorithm extracts a variety of features that are then used to train a duration-dependent hidden semi-Markov model to segment the PCG. To ease the segmentation process, some researchers apply denoising techniques to remove environmental sounds and the noises caused by the human body itself. The next phase in the ML pipeline is the feature extraction, as the features are the basis for a successful classification. Most researchers focus mainly on time, frequency, and statistical features. The widely used features are as follows: heart rate, duration of S1, S2, SYS or DIA, total power of the PCG signal, zero crossing-rate, Mel-frequency Cepstral Coefficients, Wavelet Transform, Linear Predictive Coefficients, and Shannon entropy [18], [19].

The final phase in the ML pipeline is learning and evaluation of the ML models. The most systematic comparison of the ML models was performed via the PhysioNet challenge [1]. The challenge aimed to encourage the development of algorithms to classify heart sound recordings collected from a variety of clinical or nonclinical (such as in-home visits) environments. More details about the challenge dataset can be found in section 2.2 of the *PhysioNet datasets*. During the challenge, the ML models were ranked using an average of the weighted-sensitivity and the weighted-specificity scores achieved by the models. The weights were used as a normalization factor for the noisy recordings in the data. The best score of 86.0 was achieved by Potes *et al.* [20] using a method that was based on an ensemble of classifiers combining the outputs of an AdaBoost classifier and a Convolutional Neural Network (CNN). The second-best result, which was 85.9, was achieved by Zabihi *et al.* [21] using an ensemble of feature-based feedforward neural networks. Similarly, Kay and Agarwal [22] used a fully connected, neural network and achieved a score of 85.2. In fourth place, Bobillo [23] achieved a score of 84.5 using a tensor technique. Homsi *et al.* [18], who achieved a score of 84.5, introduced an approach using a nested ensemble of algorithms that includes Random Forest, LogitBoost and a Cost-Sensitive Classifier. A probabilistic approach based on logical rules and a probability assessment was proposed by Plesinger *et al.* [19], which achieved a score of 84.1. Rubin *et al.* [24], who achieved a score of 84.0, used CNNs on MFCC heat maps.

Although the challenge datasets present a great opportunity to compare methods for classifying heart sounds, unfortunately, the challenge did not specifically include recordings from CHF patients, and second, it does not provide full access to the challenge test datasets. Thus, we cannot make any CHF-detection comparison. However, we used the publicly available challenge datasets¹ for evaluation and compared the results to the challenge baseline method [3].

¹<https://www.physionet.org/content/challenge-2016/1.0.0/>

With respect to the related work, our approach differs in the following aspects. (i) Most of the approaches from the PhysioNet challenge used the algorithm developed by Schmid *et al.* [16] for the initial segmentation of the PCG signals, which is based on the detection of the typical heart sounds. However, in noisy environments, the detection of these sounds may be even more challenging than the classification itself. For that reason, our method does not use such a strict segmentation, but rather an overlapping-sliding window technique in combination with a *segment-based* classifier. We present the analysis of the method's performance with respect to the segment (window) size. (ii) The proposed end-to-end DL architecture learns both from the temporal representation of the signal and the spectral representation of the signal, whereas most of the approaches in the related works use end-to-end learning in one of the domains only (either spectral or temporal [14]). (iii) We used the PhysioNet Challenge datasets to evaluate our approach and to provide a comparison with the challenge baseline method; additionally, we used our own dataset, which, in addition to including the typical healthy vs. patient labels, is also labeled for the specific CHF phase, i.e., compensated (when the patient feels well) and decompensated (when the patient does not feel well) for some of the patients. This allowed us to extend the study beyond the typical healthy vs. patient analysis and to explore personalized models for detecting the different CHF phases. To the best of our knowledge, this is the first computer-science study developing CHF detection models.

III. DATA DESCRIPTION

A. UKC-JSI DATASET

Seven datasets were used in our study. The first was collected by the authors of this paper, while the rest of the datasets originated from the 2016's PhysioNet challenge [1]. Our dataset (UKC-JSI, Table 1) was obtained using a professional digital stethoscope 3M™ Littmann Electronic Stethoscope Model 3200. The stethoscope allows the recording of up to 12 clips of up to 30 s in length, with a sampling rate of 4 KHz. The device uses built-in filters to reduce the ambient noise and allows different settings to focus on listening to heart or lungs – for our experiments, we always opted for the option with the minimal filtering. The recordings were transferred to a computer via a Bluetooth connection and were analyzed offline.

The study was approved by the medical ethics committee beforehand. We recorded 110 healthy people (meaning that they had no medical condition) and 51 people diagnosed with CHF. For 22 CHF patients, recordings were obtained both during the decompensation episode (when hospitalized) and during the compensated phase (when discharged).

The recordings were always obtained at Erb's point, and each recording was up to 30 s long (stethoscope's limit). For some healthy people, more than one recording was obtained to increase the amount of data in the study (recordings of patients were obtained in clinical settings, which limited the available time).

TABLE 1. Overview of our experimental data recorded on healthy individuals and on patients in decompensated and recompensated CHF episodes.

	Decomp.	Recomp.	Healthy	All
# Subjects	51	22	110	183
# Recordings	52	22	159	233
Duration (min)	17	7	52	76

TABLE 2. Overview of the six (A to F) PhysioNet Challenge datasets.

Data	#Subjects	#Recordings	Proportion (%)		
			Abnorm.	Normal	Unsure
A	121	409	67.5	28.4	4.2
B	106	490	14.9	60.2	24.9
C	31	31	64.5	22.6	12.9
D	38	55	47.3	47.3	5.5
E	356	2054	7.1	86.7	6.2
F	112	114	27.2	68.4	4.4
ALL	764	3153	18.1	73.0	9.7

B. PhysioNet DATASET

The PhysioNet 2016 Cardiology Challenge database consists of six datasets (A through F, recorded by six research groups - participants of the competition) containing a total of 3,153 heart sound recordings, lasting from 5 seconds to just over 2 minutes. The recordings were obtained in either a clinical or a nonclinical environment, from both healthy subjects and pathological patients, and from different locations on the body. The four typical locations were the aortic area, the pulmonic area, the tricuspid area, and the mitral area, but could be one of nine different locations. Both the healthy subject and pathological patient samples include both children and adults. Each subject/patient may have contributed between one and six recordings. However, there is no information in the dataset regarding which recording belongs to whom. All recordings were resampled to 2,000 Hz and stored in the “wav” format. Each recording contains only one PCG lead. As the recordings were often collected in uncontrolled environments, many of the recordings are corrupted by various noise sources, such as talking, stethoscope motion, breathing, and intestinal sounds. In addition, some recordings were difficult or even impossible to classify as normal or abnormal. The summary of the recordings in the Challenge database can be found in Table 2. In addition, each of the six teams recording the training sets also produced a testing set, which was used for the evaluations of the contributed algorithms. We did not have access to the testing sets; therefore, we do not include them in our analysis.

IV. METHOD

The outline of our method is presented in Fig. 2. It consists of the following two main components: a classic ML component (represented with colored squares on the right side of the

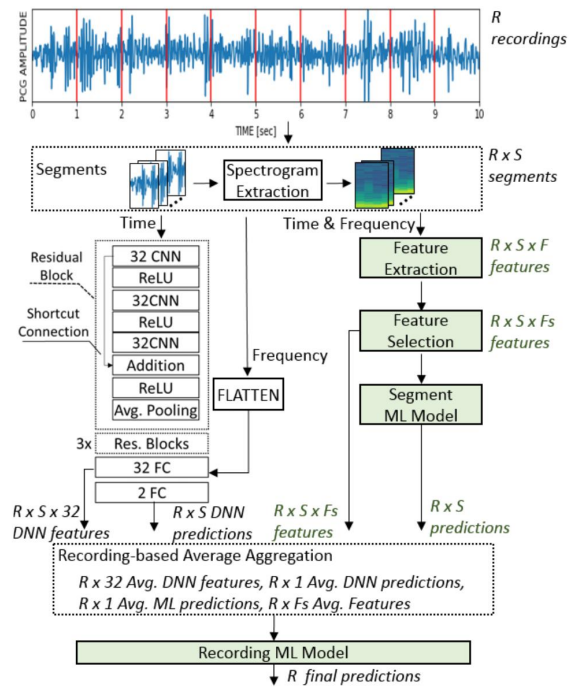
**FIGURE 2.** Proposed method. End-to-end DL (uncolored squares on the left). Classic ML (colored squares).

figure) and an end-to-end DL component (represented with noncolored squares). The input to the classic ML pipeline is the same as the input to the end-to-end DL pipeline, but the classic ML pipeline contains a feature extraction process to extract features from the raw data and to format the data into a classic ML format. The end-to-end DL does not require feature extraction, and it works directly with the raw data. Additionally, both pipelines work with signals from the time and frequency domains. The outputs of the two components are then merged by a *recording-based* ML model that outputs the final prediction, i.e., whether a recording comes from a healthy subject or from a patient. The details of the method are explained in the following subsections.

A. CLASSIC MACHINE LEARNING

The classic ML component consists of feature extraction, feature selection, and a *segment-based* ML model. As noises not related to heart sounds are expected to be present in the recordings, the first step is filtering. Most cardiovascular sounds are most likely to occur in the frequency range below 1 kHz [25]; thus, we applied a low-pass Butterworth filter with a threshold of 1 kHz to the raw audio files. For the segmentation of the filtered audio signal, we used a sliding window technique.

Next, the audio features were extracted using the OpenSMILE feature extraction tool [26]. The tool was originally created for acoustic emotion recognition in 2009 but was later expanded to more general uses [27]. For example,

in addition to affect recognition, it is widely used for music information retrieval (e.g., chord labeling, beat tracking, etc.).

Based on the related work [18], [20], [21], we extracted features from the recordings in the time and frequency domains. For example, Potes *et al.* [20] extracted features in the time and frequency domains. The time-domain features include the mean, variance, skewness, kurtosis, etc. The frequency domain features include the power across different frequency bands, 13 MFCCs, etc. Similarly, Zabihi *et al.* [21] used features in both the time and frequency domains. Their features include the following: Linear Predictive Coefficient (LPC), Entropy-based features, MFCCs, features extracted over power spectral density, etc. In addition to the typical time- and frequency-based features, Homsy *et al.* [18] added entropy and the Zero Crossing Rate (ZCR). Additionally, we included features that describe the noise in the signal, e.g., jitter, shimmer and the Harmonics-to-Noise Ratio.

The complete list of features is described by Florian and Schuller *et al.* [26], [28], [29]. For each segment, 1941 features were extracted on a segment level and 1941 additional features were extracted on a recording-level. Thus, each segment is represented by segment-based features, which describe only the segment itself, and recording-based features, which describe the whole recording. The shortest segment size used in our experiments was one second so that each segment contained at least one complete heartbeat (with common heart rates above 60 beats per minute). By extracting features both from the segments and from the whole recordings, we aim to capture both short-term information and the long-term information. The short-term information from the segment-based features mainly describes the heart beats, and the long-term information describes the overall recording. The output of the feature extraction component is marked as “ $R \times S \times F$ features” in Fig. 2, since we obtain F -features for each R -recording and S -segment.

As the number of available features for each segment (1941×2) is several times larger than the number of available recordings in most of the experimental datasets, a feature selection step is required to avoid overfitting. In general, the feature selection methods can be divided into wrapper methods, ranking methods (also known as filter methods) and a combination of the two. The wrapper methods (e.g., based on ROC [30], Bhattacharyya distance [31], etc.) produce better results compared to the ranking methods (e.g., methods based on mutual information [32], information entropy [33], Fisher score [34], Wilcoxon signed-rank test [35], etc.), but they induce a heavy computation burden. We decided to use mutual information because it is very efficient and fast to compute. However, any other feature selection or dimensionality reduction methods (e.g., PCA [36]) can be used.

Mutual information is a measure that estimates the dependency between two random variables. We ranked the features using mutual information values between the features and the class values. We used only the top-ranked m features, where m was set to be equal to the number of training samples. The features were ranked using 10% of the instances from the

training data, which were randomly selected. The output of the feature selection component is marked as “ $R \times S \times F_s$ features” in Fig. 2.

After the feature selection, a *segment-based* classifier was trained. The *segment-based* classifier uses the segments as input instances, represented via the selected features from the previous step, and outputs the estimated class probabilities for each segment (segment-class probabilities) of each recording. The output probabilities are marked as “ $R \times S$ predictions” in Fig. 2. The segment-class probabilities are later used as the input to the *recording-based* classifier. The *segment-based* classifier was trained using the Random Forest (RF) algorithm. We chose the RF algorithm because it is robust to noise in the input features. The *recording-based* classifier is described in the *Combining Classic ML and end-to end DL* section.

B. DEEP LEARNING

DL represents a class of ML algorithms that use a cascade of multiple layers of nonlinear processing units [37]. The first layer receives the input data, and each successive layer uses the output from the previous layer as input. DL architectures are able to solve complicated AI tasks (e.g., in computer vision, language, biomedicine, etc.) by learning high-level abstractions from raw data [38].

A cascade of multiple layers of nonlinear processing units, where each processing unit receives input from the previous layer, is called a Fully Connected Neural Network (FCNN). In a typical FCNN, layer i computes an output vector z_i using the following equation:

$$z_i = f(b_i + W_i z_{i-1}) \quad (1)$$

where b_i (biases) and W_i (weights) are the parameters for the i^{th} layer. z_{i-1} is the output vector of the previous layer, and z_0 is the input data. The activation function f can be a Rectified Linear Unit (ReLU), as follows [39]:

$$f(c) = \max(0, c) \quad (2)$$

or some other nonlinear function, for example, sigmoid or tanh. For classification problems, the final output layer (z_i) usually uses a softmax activation function, as follows:

$$z_{ij} = \frac{e^{b_{ij} + W_{ij} z_{i-1}}}{\sum_j e^{b_{ij} + W_{ij} z_{i-1}}} \quad (3)$$

where j represents the j^{th} row of the weights W_i . The softmax function has a nice property, as follows:

$$\sum_j z_{ij} = 1 \quad (4)$$

and it is always positive; thus, it can be used as an estimator for an input data sample x to belong to the j^{th} class for a specific problem, as follows:

$$P(y = j | x) \quad (5)$$

The parameters of the network (b_i and W_i) are learned using an optimization algorithm, for example, gradient

descent [40]. For a binary classification problem (e.g., y is either 1 or 0), binary cross-entropy is used as a loss function that is minimized over the N pairs of data samples and labels (x_n, y_n) .

$$J(\mathbf{W}) = -\frac{1}{N} \sum_n [y_n \log(p_n) + (1 - y_n) \log(1 - p_n)] \quad (6)$$

where, p_n is the estimated probability for the N^{th} sample to belong to class 1.

CNNs are a type of NN that are designed with three main architectural ideas to ensure some degree of shift, scale, and distortion invariance. This is achieved by utilizing the following: (i) local receptive fields, i.e., each unit in a layer receives input from a set of neighboring units in the previous layers; (ii) shared weights (units in a layer are organized in groups and all units in one group share the same set of weights); the set of outputs of the units in one group is called a feature map, and the set of connection weights used by the units to create the feature map is called a kernel or filter [41], [42]; (iii) spatial or temporal sampling, where, if the input is shifted, the feature map output will also be shifted [43]. In addition, because of the specific architecture (parameter sharing and local connections), the CNNs have much fewer connections and parameters to train, while their theoretical-best performance is likely to be only slightly worse compared to that of FCNNs [5].

The DNN architecture used in this study is deep *Spectro-temporal ResNet*. A similar *Spectro-temporal ResNet* architecture has already proved successful for human activity recognition in our previous study [14] by achieving comparable accuracy to state-of-the-art feature-based models. The structure is based on an idea for training very deep end-to-end networks for image recognition; i.e., it uses shortcut (residual) connections to fight the gradient-vanishing problem [44]. Additionally, our architecture consists of two branches, i.e., one that works with the raw PCG signal in the time domain and another that works with the spectral representation of the signal. The temporal information is extracted by residual blocks that contain 1D CNN filters. To reduce the internal covariance shift, each CNN layer is followed by a batch normalization layer [45]. To speed up the training process, ReLU activation layers are used [46]. For the dimensionality reduction, each residual block ends up with a maximum pooling layer. The network obtains the spectral information by calculating a spectrogram of the input signal. Toward the end of the network, the two branches, i.e., the spectral and the temporal branch, are merged using FC layers. The output of the end-to-end DL component is marked as " $R \times S$ DNN predictions", as it outputs one prediction for each segment (S) of each recording (R). Additionally, the output of the second to last FC layer is marked as " $R \times S \times 32$ DNN features", as it contains 32 hidden units; thus, it outputs a vector with a size of 32 for each segment. This vector represents the spectro-temporal encoding of each segment and is also the input to the *recording-based* ML model.

The FC layer learns a spectro-temporal encoding, which is further utilized by the *recording-based* ML model. We used two FC layers, each with a size of 32 units; thus, the spectro-temporal encoding is represented by a vector of a size of 32.

The *Spectro-Temporal ResNet* was trained by minimizing the binary cross-entropy loss function using the Adam optimizer with a learning rate of 10^{-3} and a decay of 10^{-3} . The batch size was set to 256, and the maximum number of training epochs was set to 20. The network parameters, including the number of residual blocks, the number of CNN layers per block, the size of the CNN filters, the learning rate, and the batch size, were determined experimentally.

C. COMBINING CLASSIC MACHINE-LEARNING AND END-TO-END DEEP LEARNING

The four outputs of the components before the recording-based ML model (" $R \times S \times Fs$ features", " $R \times S$ predictions", " $R \times S \times 32$ DNN features" and " $R \times S$ DNN predictions") are first averaged for each recording (thus, we obtain the averaged " $R \times Fs$ features", " R predictions", " $R \times 32$ DNN features" and " R DNN predictions") and are then used as the input to the recording-based ML model. Finally, the *recording-based* ML model outputs the final prediction for each recording. The motivation here is the fact that all the segments in a chosen recording belong to the same class. However, some segments may be more informative than others; therefore, a segment-based classification followed by the aggregation of segment-based forecasts should improve the overall classification. The *recording-based* classifier was trained using the RF algorithm.

Each of the three classifiers (the *segment-based* classifier, the *Spectro-temporal ResNet* and the *recording-based* classifier) were trained separately on the training data. Since the *recording-based* classifier is a meta-learner that utilizes the output of the *segment-based* classifier and the *Spectro-temporal ResNet*, a holdout set is required for its training. More specifically, we used a 10-fold cross-validation process on the method's training data to train the ML and DL segments. During the 10-fold cross-validation process, both the *segment-based* classifier and the *Spectro-temporal ResNet* provided output for each of the 10 folds. The recording ML was trained on those outputs. Note that once trained, the whole method was evaluated on a separate test set, which was not used in the training phase of the three ML models.

D. BASELINE METHOD

The baseline method starts with heart sound segmentation using the Springer's algorithm [17] for detecting the four states, i.e., S1, systole, S2, and diastole (see Fig. 1). After that, it extracts twenty features from the position information of the four states mentioned above. The first ten features are defined as the averages and standard deviations of the beat-to-beat intervals (RR intervals), S1, S2, systolic and diastolic intervals. The last ten features describe the averages and standard deviations of the ratios between the different intervals and the ratios between the mean absolute amplitude

during systole or diastole to that during the S1 or S2 period in each heartbeat. The features are fed to a binary logistic regression classifier. To provide a better comparison, we also used an ensemble method (Random Forest) as another baseline classifier.

V. EXPERIMENTS

We performed three types of experiments. In the first set of experiments, i.e., the *PhysioNet experiments*, we evaluated the method for classifying healthy vs. patient recordings on seven datasets and compared the results against the baseline methods. In the second set of experiments, i.e., the *UKC-JSI experiments*, we evaluated the method for different window sizes and performed subject-independent evaluation for the classification of healthy vs. patient recordings. In the third set of experiments, i.e., the personalization experiments, we analyzed the expert-features for the classification of recompensated vs. decompensated recordings on the UKC-JSI dataset and performed a subject-independent evaluation of a model built with those features. In these experiments, we used only the expert-features, as the number of recompensated/ decompensated recordings is fairly small (44) for training an end-to-end DL model. For the evaluation of the ML models, we used the following evaluation metrics (in percent):

$$\text{Accuracy} = 100 * \frac{\text{Correctly classified recordings}}{\text{Number Recordings}} \quad (7)$$

$$\text{Sensitivity} = 100 * \frac{\text{Correctly classified Patient recordings}}{\text{Patient recordings}} \quad (8)$$

$$\text{Specificity} = 100 * \frac{\text{Correctly classified Healthy recordings}}{\text{Healthy recordings}} \quad (9)$$

$$\text{Score} = \text{Average}(\text{Sensitivity}, \text{Specificity}) \quad (10)$$

A. PhysioNet EXPERIMENTS

In these experiments, we used the same evaluation approach that the organizers of the PhysioNet Challenge used to evaluate their baseline model [3]. The evaluation procedure is a recording-independent 10-fold cross-validation process, which means that one recording can be either only in the training dataset or only in the testing dataset. We performed the 10-fold evaluation for each dataset specifically. The folds were not randomized to ensure that each recording is used as a test recording at least once. The 10-fold evaluation was performed 10 times for each dataset, and the average accuracy for each dataset is presented in Table 3. For each of these datasets, our method – the *recording-based* classifier (*Rec.* in Table 3) – outperformed the other classifiers (the majority classifier, the *baseline LogReg* classifier, the *baseline RF* classifier and the *segment-based* classifier (*Seg.* in Table 3). For the *segment-based* classifier, the prediction for a recording is calculated as a majority of the predictions of its segments. Additionally, the accuracy highly depends on the dataset.

TABLE 3. Accuracy for each of the PhysioNet Challenge datasets (A-F) and for our dataset UKC-JSI (U).

	A	B	C	D	E	F	U
<i>Majority</i>	71.4	78.8	77.4	50.9	92.4	70.2	66.6
<i>LogReg</i>	72.4	79.6	77.4	67.3	92.6	72.8	64.1
<i>RF</i>	69.4	77.6	87.1	72.7	92.9	74.6	74.2
<i>Seg.</i>	77.6	80.9	90.3	89	99.7	72.7	76.8
<i>Rec.</i>	80.4	82.5	100	92.6	99.6	77.6	84.6

TABLE 4. p-values from the McNemar's tests between our method (Segment-based) and the other methods.

	<i>Segment-based</i>						
	A	B	C	D	E	F	U
<i>LogReg</i>	0.028	0.000	0.000	0.002	0.038	0.053	0.000
<i>RF</i>	0.024	0.004	0.004	0.031	0.043	0.167	0.004
<i>Seg.</i>	0.043	0.265	0.045	0.375	0.465	0.013	0.005

More specifically, the *recording-based* classifier achieved the lowest accuracy, i.e., 77.6%, for dataset F and the highest accuracy, i.e., 99.6%, for dataset E. For our dataset, the hboxUKC-JSI dataset (U in Table 3), the accuracy is 84.6%.

It is interesting to note that, for the datasets C, D, and F, the number of participants is almost equal to the number of recordings (see Table 2); thus, the evaluation is almost the same as a person-independent 10-fold cross-validation process (one person can be either only in the training dataset or only in the testing dataset). This means that for these datasets, the models are roughly person-independent.

The results presented in Table 3 show that our method achieves the highest accuracy, which indicates the practical significance of the results. To check for statistical significance, we used the McNemar's statistical test, which is a recommended statistical test for comparing two classifiers over one dataset because it has a low type I error [47], [48]. For each dataset, we performed pairwise comparisons between our method and each of the other methods. More specifically, the test compares the prediction errors made by both models and checks whether there is a significant difference between them. The p-values of these tests are presented in Table 4. It can be observed that the differences between our method and the *baseline RF* and *baseline LogReg* are statistically significant ($p < 0.05$) for six out of the seven datasets. When compared to the *recording-based* method, the differences are significant for four out of the seven datasets, which is expected because the *recording-based* classifier is one module of our method (the *recording-based* method). Most importantly, for the UKC-JSI dataset (U in Table 4), there are statistically significant differences between our method and all of the comparison methods (with p-values of 0.005, 0.004 and 0.000).

Table 5 presents a normalized confusion matrix, i.e., the rows sum up to 100, as well as the sensitivity, specificity, and score for the UKC-JSI dataset using the *recording-based* classifier and a recording-independent 10-fold cross-validation process.

TABLE 5. Normalized confusion matrix (first two rows and columns), sensitivity, specificity, and score (average value of the sensitivity and specificity) for the UKC-JSI dataset using the recording-based classifier recording-independent 10-fold cross-validation.

	Healthy	Patient	Spec.	Sens.	Score	Acc
Healthy	93.5	6.5				
Patient	33.7	66.3	93.5	66.3	79.9	84.2

TABLE 6. Aggregated confusion matrix (first two rows and columns) over all datasets; sensitivity, specificity and score (average value of the sensitivity and specificity).

		Healthy (H)	Patient (P)	Spec.	Sens.	Score	Acc
<i>LogReg</i>	H	93.1	6.9	93.1	50.1	72.0	83.1
	P	49.9	50.1				
<i>RF</i>	H	93.6	6.4	93.6	66.8	80.2	86.7
	P	33.2	66.8				
<i>Seg.</i>	H	56.7	43.3	56.7	73.7	65.2	60.8
	P	26.3	73.7				
<i>Rec.</i>	H	96.2	3.8	96.2	82.3	89.3	92.9
	P	1	82.3				

TABLE 7. Accuracy for the recording-based classifier for varying window sizes on the UKC-JSI dataset.

Window size in seconds					
10	8	6	4	2	1
82.6	81.6	82.9	84.9	86.1	84.6

The results suggest that the sensitivity is lower than the specificity, which means that the classifier produces more false negatives in these experiments.

To present a clearer comparison with the scoring method from the PhysioNet Challenge, in Table 6 we present the aggregated (and normalized) confusion matrix over all the datasets for each classifier, and the achieved sensitivity, specificity, and score. Again, it can be observed that the *recording-based* classifier is the best performing classifier, with a score of 89.3 and an accuracy of 92.9%.

B. UKC-JSI EXPERIMENTS

In these experiments, we analyzed the relation between the window size and the performance of the *recording-based* classifier. For that reason, the experiments were performed using varying window sizes (10, 8, ..., 2, and 1 second) and a 50% overlap. One second was chosen as the shortest window so that each segment contains at least one complete heart-beat (with common heart rates above 60 beats per minute). The evaluation was performed using the same recording-independent 10-fold cross-validation process as before to obtain comparable results. We only used the UKC-JSI dataset for these experiments, as our goal is to tune the algorithm for this dataset. The results are presented in Table 7.

Finally, the best performing classifier – the *recording-based* classifier using a sliding window of 2 seconds and a 50% overlap – was evaluated using a Leave-One-Subject-Out

TABLE 8. Normalized confusion matrix (first two rows and columns), sensitivity, specificity, and score (average values of the sensitivity and specificity) for the UKC-JSI dataset using the recording-based classifier and a leave-one-subject-out evaluation.

	Healthy	Patient	Spec.	Sens.	Score	Acc
Healthy	94.6	3.4				
Patient	28.5	71.5	94.6	71.5	83.0	86.3

(LOSO) evaluation procedure. This evaluation produces more a reliable estimate of the method's performance for a new (unseen) subject. The LOSO evaluation is a type of person-independent k-fold cross-validation, where k is equal to the number of participants in the dataset. We could carry out the LOSO experiment only on the UKC-JSI dataset since the PhysioNet dataset does not specify which recording belongs to which individual. The results are presented in Table 8. The accuracy of 86.3% is similar to the accuracy from the previous experiments (84.6%), which confirms that overfitting was avoided in the previous evaluation. Similarly, as in previous experiments, the sensitivity is lower than the specificity.

C. PERSONALIZATION

For 22 out of 51 patients in the UKC-JSI dataset, we obtained one recording in the decompensated phase, i.e., when the patient needs medical attention, namely, at the time of hospital admission, and one in the recompensated phase, i.e., when the patient feels well and is released from the hospital. We analyzed whether we can distinguish between the two phases, as this is the first step toward the long-term goal of building a model that would allow us to continuously monitor the worsening of the CHF condition.

We performed statistical tests to check whether there is a difference in the features when calculated from the recompensated recordings compared to the decompensated recordings. We used the Wilcoxon signed-rank test, a nonparametric statistical hypothesis test used to determine whether two paired samples are sampled from the same distribution [49]. In this experimental setting, one sample contains the values of a specific feature extracted from the recompensated recordings and the other sample contains the values for the same feature but extracted from the decompensated recordings. The informative features should have different distributions depending on the type of recording. Table 9 presents the 15 features for which the tests showed p-values smaller than 0.001.

Finally, we built a transparent ML model that can distinguish between the two phases. Since the dataset, which contains 44 recordings from 22 patients, is too small, we used a simple decision tree classifier conditioned to a maximum depth of two, to minimize overfitting. For training the model, we used the features in Table 9.

Before feeding them to the algorithm, the features were normalized by subtracting the person-specific mean value from each feature. Table 10 presents the evaluation results (normalized confusion matrix, sensitivity, specificity, score, and accuracy). Only 3 out of 44 recordings were

Spectro-Temporal model avoids overfitting better than the Temporal model.

Regarding the evaluation of the overall method, the aggregated results (Table 6) showed that the method achieved an overall accuracy of 92.9% and a score of 89.3, which represents an improvement compared to the results achieved by the *baseline RF* (accuracy of 86.7% and score of 80.2) and *baseline LogReg* (accuracy of 83.1% and score of 70.2). Additionally, our method's accuracy of 92.9% (or the method's error of 7.1%) is quite close to the percentage of recordings labeled as "unknown" by experts (see Table 2), with the rest of the recordings labeled as healthy or unhealthy. This is another indicator that our method performs well for heart-sound classification. Most importantly, our method achieved a 10% higher accuracy compared to the *baseline RF* for the CHF domain (see the results in Table 3 for the dataset U – UKC-JSI).

The windowing analysis in the *UKC-JSI experiments* section showed that for the UKC-JSI dataset, the method achieves an accuracy higher than 81% for all of the tested window sizes. The accuracy achieved by the *baseline RF* on the same dataset was 74.2%, and the accuracy achieved by the *baseline LogReg* was 64.1% (see Table 3). Thus, our method outperforms the baseline methods for any window size, indicating that the method is quite robust with respect to the window size, which was also supported by the LOSO evaluation (see Table 8). It seems that better accuracy is achieved for smaller rather than larger window sizes (see Table 7).

Regarding the related work, one very recent study for measuring the timing of heart sound components through PCG data was presented by Giordano and Knaflitz [50]. Their study focuses on the detection of the typical S1, S2 and S3 heart sounds. A very similar approach was presented by Tseng et al. [51]. These approaches depend heavily on the detection of the typical S3 sound, which is mostly conditioned by the quality of the PCG signal. A direct comparison with the related work is presented in Table 11. The baseline methods in this study (*baseline RF* and *baseline LogReg*) use features that are based exactly on the detection of the typical heart sounds [16]. The evaluation results showed that these methods perform poorly compared to our method. The main reason for the poor performance is the noise in the data. More general ML approaches for healthy vs. unhealthy heart sound classification are the methods from the PhysioNet challenge. The winning algorithm of the challenge achieved a score of 85 in their internal evaluation [20]. For the evaluation, they used 20% of the PhysioNet data, which was randomly selected. Our method achieves a score of 89 using a 10-fold cross-validation process, which is basically 10 evaluations over 10% of the data, which was randomly selected. Although the results are not directly comparable, the scores indicate that our method performs quite well compared to the state-of-the-art methods.

Additionally, our main goal was not to create a general method for healthy vs. unhealthy heart-sound detection, but

TABLE 11. Comparison with the related work.

	Method	Score
Clifford et al. [3]	Logistic Regression (<i>baseline LogReg</i>)	72
Clifford et al. [3]	Random Forest (<i>baseline RF</i>)	80
Potes et al. [20]	AdaBoost (features) + CNN (end-to-end)	85
Our method	Random Forest (features) + Spectro-temporal ResNet (end-to-end)	89

rather, our focus is on CHF detection. In addition to the studies that use the detection of the typical heart sounds (S1, S2, S3), we were not able to find any studies on CHF using machine learning except for our previous work in this field [1], [53]. One very recent study on CHF detection was presented by Porumb et al. [52], where they used CNNs on data collected with ECG devices. Our work differs from theirs, as we used PCG.

The feature selection step of the proposed method can significantly influence the classification performance of the feature-based ML methods. In the future, we will conduct an extensive feature selection study to find the optimal feature subset. Additionally, in this work, we tested dataset-specific models. In the future, one could employ transfer-learning techniques [54] to utilize data from other similar studies/datasets. We envision a system capable of detecting new CHF patients by using the proposed method in this study. Once detected, our initial analysis showed that it is relatively simple to build personalized models (e.g., by using some of the identified features in this study) for monitoring the different CHF phases. In the long term, our approach would allow patients to actively cooperate in the process and the treating physicians would be able to adjust patients' medical management in a timely manner and thus avoid hospital admissions.

VII. CONCLUSION

In this paper, we presented a novel method for CHF detection from PCG audio recordings. The method combines classic ML and end-to-end DL. The classical ML learns from a large body of expert-defined features and the DL learns both from the time-domain (i.e., the raw PCG signal) representation of the signal and the spectral representation of the signal. We evaluated the method on our own dataset for CHF detection and additionally on six publicly available PhysioNet datasets used for the recent PhysioNet Cardiology Challenge. The challenge datasets allowed us to extensively evaluate the performance of the method on similar domains. The evaluation results on all the datasets showed that, compared to the challenge baseline methods, our method achieves the best performance (see the *PhysioNet experiments* section). The facts that most of these datasets are labeled for different types of heart-related conditions and that the PCG audio is recorded from a different body position in most of the datasets (e.g., aortic area, pulmonic area, tricuspid area, and mitral area) strongly indicate that the proposed method is quite robust and that it is useful for detecting different

types of heart-sound classification problems and not just for CHF detection, as long as domain-specific labeled data are provided.

Finally, we extended the study beyond the typical healthy vs. patient classification and explored personalized models for detecting different CHF phases, i.e., the recompensated phase (i.e., when the patient feels well) and the decompensated phase (i.e., when the patient needs medical attention). We identified 15 features that have different distributions depending on the phase. By using just two of these features, we were able to build a simple and transparent decision tree classifier (see Fig. 3) that is capable of distinguishing between the recompensated and the decompensated phases with an accuracy of 93.2%, calculated using a LOSO evaluation. While we are aware that there is a risk of overfitting in these final experiments, especially since the dataset contains only 44 samples, we believe that these results are very encouraging and represent a solid base for further development of personalized models. To the best of our knowledge, this is the first study to address such a problem.

ACKNOWLEDGMENT

The authors would like to thank A. Gradišek, MD, who first brought to their attention the idea of detecting CHF based on heart sounds.

REFERENCES

- [1] M. Gjoreski, M. Simjanoska, A. Gradišek, A. Peterlin, M. Gams, and G. Poglajen, "Chronic heart failure detection from heart sounds using a stack of machine-learning classifiers," in *Proc. Int. Conf. Intell. Environ. (IE)*, Aug. 2017, pp. 14–19.
- [2] J. Voigt, M. S. John, A. Taylor, M. Krucoff, M. R. Reynolds, and C. M. Gibson, "A reevaluation of the costs of heart failure and its implications for allocation of health resources in the United States," *Clin. Cardiol.*, vol. 37, no. 5, pp. 312–321, May 2014.
- [3] G. D. Clifford, C. Liu, B. Moody, D. Springer, I. Silva, Q. Li, and R. G. Mark, "Classification of normal/abnormal heart sound recordings: The PhysioNet/computing in cardiology challenge 2016," in *Proc. Comput. Cardiol. Conf. (CinC)*, May 2017, pp. 609–612.
- [4] X. Jiang, Y. Pang, X. Li, and J. Pan, "Speed up deep neural network based pedestrian detection by sharing features across multi-scale models," *Neurocomputing*, vol. 185, pp. 163–170, Apr. 2016.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017.
- [7] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.
- [8] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.
- [9] S. Amiriparian, A. Baird, S. Julka, A. Alcorn, S. Ottl, S. Petrovic, E. Ainger, N. Cummins, and B. W. Schuller, "Recognition of echolalic autistic child vocalisations utilising convolutional recurrent neural networks," in *Proc. INTERSPEECH*, Aug. 2018, pp. 2334–2338.
- [10] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, S. Pugachevskiy, and B. Schuller, "Bag-of-deep-features: Noise-robust deep feature representations for audio analysis," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.
- [11] M. Gjoreski, H. Gjoreski, M. Luštrek, and M. Gams, "Deep affect recognition from R-R intervals," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput. ACM Int. Symp. Wearable Comput. (UbiComp)*, 2017, pp. 754–762.
- [12] H. P. Martinez, Y. Bengio, and G. N. Yannakakis, "Learning deep physiological models of affect," *IEEE Comput. Intell. Mag.*, vol. 8, no. 2, pp. 20–33, May 2013.
- [13] M. Gams, "Weak intelligence: Through the principle and paradox of multiple knowledge," in *Advances in Computation: Theory and Practice*, vol. 6. Hauppauge, NY, USA: Nova Science Publishers, 2001, p. 245.
- [14] M. Gjoreski, V. Janko, G. Slapnicar, M. Mlakar, N. Rescic, J. Bizjak, V. Drobnic, M. Marinko, N. Mlakar, M. Lustrek, and M. Gams, "Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors," *Inf. Fusion*, 2019.
- [15] M. Gjoreski, A. Gradišek, B. Budna, M. Gams, and G. Poglajen, "Toward early detection and monitoring of chronic heart failure using heart sounds," in *Proc. 15th Int. Conf. Intell. Environ. Workshop*, vol. 26. IOS Press, Aug. 2019, p. 336.
- [16] S. Schmidt, E. Toft, C. Holst-Hansen, C. Graff, and J. Struijk, "Segmentation of heart sound recordings from an electronic stethoscope by a duration dependent Hidden-Markov model," in *Proc. Comput. Cardiol.*, Sep. 2008, pp. 345–348.
- [17] D. B. Springer, L. Tarassenko, and G. D. Clifford, "Logistic regression-HSMM-based heart sound segmentation," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 4, pp. 822–832, Apr. 2015.
- [18] M. N. Homsí, N. Medina, M. Hernandez, N. Quintero, G. Perpiñan, A. Quintana, and P. Warrick, "Automatic heart sound recording classification using a nested set of ensemble algorithms," in *Proc. Comput. Cardiol. Conf. (CinC)*, May 2017, pp. 817–820.
- [19] F. Plesinger, I. Viscor, J. Halamek, J. Jurco, and P. Jurak, "Heart sounds analysis using probability assessment," *Physiol. Meas.*, vol. 38, no. 8, pp. 1685–1700, May 2017.
- [20] C. Potes, S. Parvaneh, A. Rahman, and B. Conroy, "Ensemble of feature: Based and deep learning: Based classifiers for detection of abnormal heart sounds," in *Proc. Comput. Cardiol. Conf. (CinC)*, May 2017, pp. 621–624.
- [21] M. Zabihi, A. Bahrami Rad, S. Kiranyaz, M. Gabbouj, and A. K. Katsaggelos, "Heart sound anomaly and quality detection using ensemble of neural networks without segmentation," in *Proc. Comput. Cardiol. Conf. (CinC)*, May 2017, pp. 613–616.
- [22] E. Kay and A. Agarwal, "DropConnected neural networks trained on time-frequency and inter-beat features for classifying heart sounds," *Physiol. Meas.*, vol. 38, no. 8, pp. 1645–1657, Jul. 2017.
- [23] I. D. Bobillo, "A tensor approach to heart sound classification," in *Proc. Comput. Cardiol. Conf. (CinC)*, May 2017, pp. 629–632.
- [24] J. Rubin, R. Abreu, A. Ganguli, S. Nelaturi, I. Matei, and K. Sricharan, "Classifying heart sound recordings using deep convolutional neural networks and mel-frequency cepstral coefficients," in *Proc. Comput. Cardiol. Conf. (CinC)*, May 2017, pp. 813–816.
- [25] S. Choi and Z. Jiang, "Comparison of envelope extraction algorithms for cardiac sound signal segmentation," *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1056–1069, Feb. 2008.
- [26] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in openSMILE, the munich open-source multimedia feature extractor," in *Proc. ACM Multimedia (MM)*, vol. 5. Barcelona, Spain: ACM, Oct. 2013, pp. 835–838.
- [27] A. Gradišek, G. Slapnicar, J. Šorn, M. Luštrek, M. Gams, and J. Grad, "Predicting species identity of bumblebees through analysis of flight buzzing sounds," *Bioacoustics*, vol. 26, no. 1, pp. 63–76, Jan. 2017.
- [28] B. Schuller, M. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pantic, "AVEC 2011—The first international audio/visual emotion challenge," in *Proc. Int. Conf. Affect. Comput. Intell. Interact. Berlin, Germany: Springer*, Oct. 2011, pp. 415–424.
- [29] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: The munich versatile and fast open-source audio feature extractor," in *Proc. 18th ACM Int. Conf. Multimedia*, 2010, pp. 1459–1462.
- [30] R. Wang and K. Tang, "Feature selection for maximizing the area under the ROC curve," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Dec. 2009, pp. 400–405.
- [31] G. Xuan, X. Zhu, P. Chai, Z. Zhang, Y. Q. Shi, and D. Fu, "Feature selection based on the bhattacharyya distance," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 4, 2006, p. 957.
- [32] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, 2004, Art. no. 066138.
- [33] C. Arndt, *Information Measures: Information and Its Description in Science and Engineering*. Springer, 2001.
- [34] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," 2012, *arXiv:1202.3725*. [Online]. Available: <https://arxiv.org/abs/1202.3725>
- [35] R. F. Woolson, *Wilcoxon Signed-Rank Test*. Hoboken, NJ, USA: Wiley, 2007, pp. 1–3.
- [36] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.

- [37] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, 2014.
- [38] Y. Bengio, "Learning deep architectures for AI," *FNT Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
- [39] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.
- [40] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.
- [41] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 609–616.
- [42] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. Lecun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2146–2153.
- [43] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [44] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," *Tech. Rep.*, 2001.
- [45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: <https://arxiv.org/abs/1502.03167>
- [46] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.
- [47] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
- [48] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998.
- [49] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics*. Springer, 1992, pp. 196–202.
- [50] N. Giordano and M. Knaflitz, "A novel method for measuring the timing of heart sound components through digital phonocardiography," *Sensors*, vol. 19, no. 8, p. 1868, Apr. 2019.
- [51] Y.-L. Tseng, P.-Y. Ko, and F.-S. Jaw, "Detection of the third and fourth heart sounds using Hilbert-Huang transform," *BioMed. Eng. OnLine*, vol. 11, no. 1, p. 8, 2012.
- [52] M. Porumb, E. Iadanza, S. Massaro, and L. Pecchia, "A convolutional neural network approach to detect congestive heart failure," *Biomed. Signal Process. Control*, vol. 55, Jan. 2020, Art. no. 101597.
- [53] M. Gjoreski, A. Gradišek, B. Budna, M. Gams, and G. Poglajen, "Toward early detection and monitoring of chronic heart failure using heart sounds," in *Proc. Intell. Environ. Workshop 15th Int. Conf. Intell. Environ.*, 2019, p. 336.
- [54] M. Gjoreski, S. Kalabakov, M. Luštrek, and H. Gjoreski, "Cross-dataset deep transfer learning for activity recognition," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput. 2019 ACM Int. Symp. Wearable Comput. (UbiComp/ISWC)*, 2019, pp. 714–718.



MARTIN GJORESKI received the B.S. degree in computer science from the Faculty of Computer Science and Engineering, Skopje, Macedonia, in 2014, and the M.S. degree in computer science from the Jozef Stefan Postgraduate School, Ljubljana, Slovenia, in 2016, where he is currently pursuing the Ph.D. degree in computer science.

He has been a Research Assistant with the Department of Intelligent Systems, Jozef Stefan Institute, Ljubljana, since 2014. His research focuses on the development of machine-learning methods for monitoring human physiological and psychological states using wearable sensors. He developed parts of the machine-learning algorithm that received the Sussex-Huawei Locomotion Challenge, in 2018 and 2019.



ANTON GRADIŠEK received the Ph.D. degree in solid-state physics from the Faculty of Mathematics and Physics, University of Ljubljana, in 2012. He worked as a postdoc at the Korea Basic Science Institute, Daejeon, South Korea, and was a Fulbright Scholar at Washington University, Saint Louis, Missouri. He is currently working at the Jozef Stefan Institute, Ljubljana, Slovenia, at the Department of Intelligent Systems and Department of Solid State Physics. His research interests include hydrogen storage materials, liquid crystals, and applications of artificial intelligence on various domains, especially medicine and bioacoustics. He was a member of a team that reached the finals of the XPrize Tricorder competition.



BORUT BUDNA received the B.Sc. degree in computer and information science from the Faculty of Computer and Information Science, University of Ljubljana, in 2017, where he is currently pursuing the M.Sc. degree. He works as a Student Researcher at the Department of Intelligent Systems, Jozef Stefan Institute. His main research interests include applications of machine learning in medicine and bioacoustics.



MATJAŽ GAMS received the Ph.D. degree. He is currently the Head of the Department of Intelligent Systems, Jozef Stefan Institute, Ljubljana, and is a Professor of computer science with the Jozef Stefan Postgraduate School and University of Ljubljana. His professional interests include intelligent systems, artificial intelligence, cognitive science, intelligent agents, electronic and mobile health, business intelligence, and information society. He is a member of several international program committees of scientific meetings, National and European strategic boards and institutions, and editorial boards of 11 journals, and he is the Managing Director of the journal *Informatica*. He is a member of the National Council of Slovenia, representing the field of science for the term of 2017–2022. His team received two activity recognition competitions and placed in the finals of the XPrize Tricorder competition.



GREGOR POGLAJEN received the M.D. and Ph.D. degrees. He is a Cardiologist staff at the Advanced Heart Failure and Transplantation Center, UMC Ljubljana, Slovenia. He is a member of a focused team, led by prof. Bojan Vrtovec, M.D., Ph.D., whose main clinical interests are advanced heart failure, heart transplantation, and mechanical circulatory support, and his main research focus is in regenerative medicine and remote patient management.

...

5.2 Driver Distractions Monitoring

Received February 28, 2020, accepted March 22, 2020, date of publication April 9, 2020, date of current version April 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2986810

Machine Learning and End-to-End Deep Learning for Monitoring Driver Distractions From Physiological and Visual Signals

MARTIN GJORESKI^{1,2}, MATJA Ž GAMS^{1,2}, (Member, IEEE),
MITJA LUŠTREK^{1,2}, (Member, IEEE), PELIN GENÇ³,
JENS-U. GARBAS³, AND TEENA HASSAN³

¹Jožef Stefan Institute, 1000 Ljubljana, Slovenia

²Jožef Stefan Postgraduate School, 1000 Ljubljana, Slovenia

³Intelligent Systems Group, Fraunhofer Institute for Integrated Circuits IIS, 91058 Erlangen, Germany

Corresponding author: Martin Gjoreski (martin.gjoreski@ijs.si)

This work was supported in part by the Slovenian Research Agency (ARRS) under Grant U2-AG-16/0672, 0287, the Fraunhofer Society via the Young Research Class 2016 program on 'Cognitive Machines' and by Fraunhofer IIS under the Affective Sensing project. The research collaboration between Jožef Stefan Institute and Fraunhofer IIS is funded by a grant received from the German Academic Exchange Service and ARRS in 2019 (Grant 57451027).

ABSTRACT It is only a matter of time until autonomous vehicles become ubiquitous; however, human driving supervision will remain a necessity for decades. To assess the driver's ability to take control over the vehicle in critical scenarios, driver distractions can be monitored using wearable sensors or sensors that are embedded in the vehicle, such as video cameras. The types of driving distractions that can be sensed with various sensors is an open research question that this study attempts to answer. This study compared data from physiological sensors (palm electrodermal activity (pEDA), heart rate and breathing rate) and visual sensors (eye tracking, pupil diameter, nasal EDA (nEDA), emotional activation and facial action units (AUs)) for the detection of four types of distractions. The dataset was collected in a previous driving simulation study. The statistical tests showed that the most informative feature/modality for detecting driver distraction depends on the type of distraction, with emotional activation and AUs being the most promising. The experimental comparison of seven classical machine learning (ML) and seven end-to-end deep learning (DL) methods, which were evaluated on a separate test set of 10 subjects, showed that when classifying windows into distracted or not distracted, the highest F1-score of 79% was realized by the extreme gradient boosting (XGB) classifier using 60-second windows of AUs as input. When classifying complete driving sessions, XGB's F1-score was 94%. The best-performing DL model was a spectro-temporal ResNet, which realized an F1-score of 75% when classifying segments and an F1-score of 87% when classifying complete driving sessions. Finally, this study identified and discussed problems, such as label jitter, scenario overfitting and unsatisfactory generalization performance, that may adversely affect related ML approaches.

INDEX TERMS Machine learning, deep learning, driver distraction, sensors, facial expressions.

I. INTRODUCTION

Every year, 25,000 people lose their lives on EU roads, and a vast majority of these accidents are caused by human errors. These errors can be avoided with advanced safety features. The monitoring of driver distractions is one such feature that can facilitate the realization of EU's long-term objective of

moving close to zero fatalities and severe injuries by 2050.¹ The transition from fully human to autonomous driving often contributes to drivers being less focused, e.g., due to drivers having the freedom to execute additional tasks or due to a potential overloading of sensory activity. On the other hand, automation levels 2 and 3 defined in SAE International's standard J3016 [1] require human attention and readiness to

The associate editor coordinating the review of this manuscript and approving it for publication was Wuliang Yin.

¹https://ec.europa.eu/commission/presscorner/detail/en/IP_19_1793

take over control in difficult situations or when the vehicle requests. Thus, the detection of distracted driving would be especially valuable in future vehicles, at least until complete autonomy is realized. For example,

Affective computing (also called artificial emotional intelligence) is the ability of technical systems to recognize and process human affective states, which can be used to enrich and facilitate human-computer interaction (HCI) [1]. The recognition of the human physical state using sensors is now mature, e.g., every mobile device is now capable of recognizing activity based on acceleration sensors. However, it is rare for a device to be capable of recognizing human mental states, e.g., stress, mental health, cognitive load, and distractions. Thus, the recognition of the human mental state is the new frontier where the most important research is being conducted. It can be used for services that are directly related to the psychological state and for enhanced HCI.

An important application of affective computing is the detection of human driver distractions [3]. Regan *et al.* [4] consider driver distraction as a subcategory of driver inattention, which is defined as “diversion of attention away from activities critical for safe driving toward a competing activity, which may result in insufficient or no attention to activities critical for safe driving” (pp. 1776). Hanowski *et al.* [5] present a list of tasks that could lead to diversion of attention. The list includes dialing and texting on the phone, reading, writing and route checking on a map [6].

Since diverted attention of the driver influences driving safety, the ability to sense the driver’s mental state is crucial [3]. These data can be gathered using sensors that are worn by the driver or by using sensors that are embedded in the car, such as video cameras. The recorded data could include behavioral cues, such as facial expressions and gestures, or physiological parameters, such as the heart rate, respiration rate and electrodermal activity. By combining this information with driving parameters and contextual information, safety risks could be estimated and timely alerts could be issued to the driver to avoid accidents and save lives.

This paper presents machine-learning (ML)-based methods for detecting driver distraction using multimodal data. The main contributions of this paper are as follows:

- Statistical analysis for the identification of the best features and modalities for detecting each of four types of distraction, namely, cognitive, emotional, sensorimotor and mixed distraction.
- Comparison of classical ML and end-to-end deep learning (DL) models for driver distraction detection, including an analysis with respect to the size of the input window and the type of the input modality: AUs, emotional activation (EMO), heart rate (HR), breathing rate (BR), nasal electrodermal activity (nEDA) or palm EDA (pEDA).
- Identification and discussion of problems such as label jitter, scenario overfitting and generalization performance that may hinder related ML approaches.

The remainder of the paper is organized as follows: Section II presents the related work. Section III describes the data that are used in this work. Section IV presents the proposed ML and DL methods. Section V elaborates the experiments and the experimental results. Section VI discusses the results, and Section VII concludes the study.

II. RELATED WORK

When analyzing systems for detecting driver distraction, one should consider the distraction types, the input signals and the detection methods. Regarding the distraction types, Gomez *et al.* [8] argued that people differ in terms of their reactions to the same distraction during driving. Thus, the relationship among the distractions, the driver reaction and, consequently, traffic accidents is complex. The distractions can occur in visual, manual or cognitive ways [9], [10]. In this study, cognitive, emotional, sensorimotor and mixed distractions are analyzed.

A. INPUT SIGNALS

The input signals can be direct, namely, measured directly from the driver, or indirect, namely, measured from the vehicle. Vehicle acceleration, steering and braking activities are examples of indirect signals of the driver’s state [1]. The related work suggests that methods that use indirect input can be informative for detecting driver distractions. Indirect detection methods rely on the vehicle behavior and are often implemented in recently produced cars. Aksjonov *et al.* [24] presented a method for detecting the driver’s distraction by monitoring lane maintenance and speed performance on specified road segments. Saito *et al.* [25] proposed an assistance system for prediction the driver’s state based on the lane departure duration. Apostoloff and Zelinsky [26] studied the driver’s attention to lane maintenance task. Castignani *et al.* [27] developed a system, namely, Sense-Fleet, that can identify risky driving events by examining the acceleration, braking and steering activities of the driver. Similarly, Pavlidis *et al.* [7] presented a statistical analysis of the relation between driver distractions and the speed, acceleration, brake force, steering and lane position. Wang *et al.* [28] proposed a forward collision warning algorithm that depends on the driver’s braking activity. However, if the vehicle is in an autonomous-driving mode, the indirect inputs will not reflect the driver’s behavior; instead, they will reflect the behavior of the algorithm for autonomous driving. Additionally, cars may be easily retrofitted with systems that use direct inputs. Thus, this study focuses on input signals that are measured directly from the driver using physiological sensors and visual analysis.

The direct input signals can be divided into two subgroups: (i) visual measurements, such as eye gaze, pupil diameter, head pose, facial expressions and driving posture, (ii) and physiological signals, such as electroencephalogram (EEG), electrooculogram (EOG), electrocardiogram (ECG), electromyogram (EMG), photoplethysmogram (PPG) and electrodermal activity (EDA) signals. The physiological

measurements provide important cues regarding the driver's state, such as his or her drowsiness and stress levels. Lee and Chung [11] evaluated eye tracking and PPG features with a dynamic Bayesian-network-based framework for the detection of driver drowsiness. Lin *et al.* [12] measured the drowsiness of the driver by using EEG signals. They decreased the number of EEG features by using the principal component analysis (PCA) method. Then, these features were fed into a linear regression model for the estimation of the drowsiness level. In addition to the EEG signals, Khushaba *et al.* [13] analyzed the drowsiness of the driver by using EOG and ECG signals. Multiple modalities, such as ECG, EMG, EDA and the respiration rate, have also been used to detect the stress level [14]. In various studies, the physiological sensors were integrated into driving equipment. For example, Singh *et al.* [15] used ECG signals that were measured via electrodes that were placed on the seat and seatbelt. Similarly, Lee *et al.* [16] measured ECG signals via electrodes that were placed on the steering wheel. Additionally, they derived the respiratory rate and HR variability from the ECG signals and used PPG that was measured from the driver's finger.

Visual measurements give the driver more freedom than physiological measurements that are obtained using wearable sensors. Bergasa *et al.* [17] measured the degree of eye closure, the eye closure duration, the blinking and nodding frequencies, and the head pose and conducted eye tracking, and they used these data to estimate the driver's state. Omidyeganeh *et al.* [18] argued that yawning is an important characteristic for estimating driver drowsiness. They used face and mouth features to detect yawning. Vicente *et al.* [19] proposed an eyes-off/on-the-road detection system that is based on head pose and eye gaze estimation. Murphy-Chutorian and Trivedi [20] argued that the driver's head pose is a strong indicator of his or her current focus of attention. Similarly, Smith *et al.* [21] analyzed the driver's attention from head- and face-related features. The hand position was also proposed as an indicator for detecting driver distraction.

In the related studies, there is no consensus regarding the input signals for the detection of driver distraction. Thus, in this study, experiments with both data from physiological sensors and data from video-based sensors were made. The physiological data include pEDA, the HR and the BR. The visual data include nEDA (extracted from data that were captured using a thermal camera), eye tracking data (x-y positions and the pupil diameter), head pose, facial expressions and emotional activation.

B. DISTRACTION DETECTION METHODS

Sikander and Anwar [29] grouped the methods for detecting driver distraction into three subgroups: mathematical models, rule-based models and models that are based on ML algorithms. Most mathematical models are designed for pre-determined setups, such as workplace and factory worker

workloads. These models consider circadian cycles, sleep history, duration of sleep and wakefulness for the detection of fatigue and performance [30]. For example, the System for Aircrew Fatigue Evaluation (SAFE) is based on such mathematical models [31]. Regarding the rule-based systems, Lee *et al.* [16] derived if-then rules and applied kernel fuzzy-C-mean to detect driving distractions. Azim *et al.* [32] proposed two-layered rule-based systems that were based on eye and mouth state information, where each layer had its own if-then rules.

The most advanced methods for monitoring driving distractions are based on ML algorithms. These methods can be classical, deep or a combination of both classical and DL [29]. Goel *et al.* [10] evaluated random forest, Naïve Bayes, SVM and decision tree for the detection of driving distraction. Random forest outperformed all the other strategies. Lee *et al.* [23] analyzed hand movements that were detected by acceleration sensors in a smartwatch. They calculated features and fed them into a support vector machine (SVM) classifier.

In addition to the classical feature-based ML methods, end-to-end DL methods, namely, methods for which feature extraction is not required and raw inputs are fed into the models, were also proposed. Masood *et al.* [6] detected distractions and causes of the distractions by using CNNs. Majdi *et al.* [34] developed Drive-Net, which combines CNN and Random Forest for the detection of the distraction categories in images. Yan *et al.* [22] used CNNs to detect various driving postures in images, such as normal driving, cell phone call, eating and smoking. Hssayeni *et al.* [35] compared two approaches for distracted driving detection: the use of traditional handcrafted image-based features along with SVM and the use of features from three end-to-end CNNs, namely, AlexNet, VGG-16 and ResNet-152. ResNet and VGG-16 outperformed AlexNet by almost 10%. The feature-based SVM realized much lower accuracy than the CNNs. Similarly, Koesdwiady *et al.* [33] used VGG-19.

All the end-to-end DL approaches use image data as input and are based on available DL architectures that have been successfully applied on images (e.g., AlexNet, VGG-16 and ResNet-152), and most focus on only one architecture. The DL architectures in this study use 1D signals as inputs; thus, specialized DL architectures for multimodal time-series data were investigated. As few studies have been conducted on end-to-end learning on 1D signals, seven DL architectures were compared in this study. To the best of our knowledge, this is the first study on the detection of driver distraction that analyzes end-to-end learning on signals using 1D convolutions and long short-term memory neural networks (LSTMs). Additionally, the DL architectures were compared to state-of-the-art classical ML algorithms using an extensive set of features. Comparison among different features/modalities for detecting driver distraction with both the classical and the DL models was also made.

TABLE 1. Layout of the driving sessions. Red represents the driving segments that include an external distraction. N represents normal driving, S represents driving under a distraction and F represents a brake failure event. In the experiments, the brake failure event is regarded as driving under a distraction.

Driving session Type	Session segment				
	1	2	3	4	5
Cognitive drive (CD) – Analytical questions	N	S	N	S	N
Emotional drive (ED) – Emotional questions	N	S	N	S	N
Sensorimotor drive (SD) – Texting on smartphone while driving	N	S	N	S	N
Normal drive (ND) – Same speed limit, heavy traffic and construction blockades	N	N	N	N	N
Relaxed drive (RD) – Same speed limit with light traffic and smooth lane changes.	N	N	N	N	N
Failure-Normal drive (FDN) – Unexpected failure in vehicle leading to a collision course.	N	N	F		
Failure-Loaded drive (FDL) – Unexpected failure in combination with mixed distractions	N	S	F		

III. DATA DESCRIPTION

The experimental data are obtained from a study by Pavlidis *et al.* [7]. In the study, they analyzed the driving behaviors of 68 volunteers in a driving simulator under a variety of distractions. Each volunteer had several driving sessions, which included a normal driving session without distractions and sessions under cognitive, emotional, sensorimotor and mixed distractions. The experimental design and the specific stressors are presented in Table 1.

Pavlidis *et al.* [7] analyzed the relations between the distractions and various driving parameters, such as the speed, acceleration, brake force, steering and lane position. From the physiological response, only nEDA [36], [37] was analyzed. In this study, the overall physiological and affective responses in relation to the external distractions were analyzed. The physiological response includes nEDA, pEDA, HR, BR and eye tracking data. The affective response includes emotions, facial expressions and the head pose.

The physiological response, which was measured using physiological sensors, and the emotional response, which was extracted from facial videos using a software that outputs probability estimates for eight prototypical emotions, were already provided in the dataset. As an addition, the facial expressions in the form of AUs and the head pose were extracted using the facial-expression-analysis software that was presented in Hassan *et al.* [38], which is hereafter referred to as AUREADER. AUREADER estimates the intensities of 22 facial action units (AUs) using a dynamic state estimation framework that fuses viscoelastic models for facial

TABLE 2. 46 channels of information that were used in the study, grouped per modality.

Modality	Channels
Physiological sensors (3)	pEDA, HR and BR
Thermal camera (1)	nEDA
Eye tracking (4)	eye position X and Y coordinates, left and right pupil diameters
Video camera (8)	emotional response: anger, contempt, disgust, fear, joy, sad, surprise, neutral
AUREADER (30)	AU01InnerBrowRaiser, AU02OuterBrowRaiser, AU04BrowLowerer, AU05UpperLidRaiser, AU06CheekRaiser, AU07LidTightener, AU09NoseWrinkler, AU10UpperLipRaiser, AU11NasolabialDeepener, AU12LipCornerPuller, AU13SharpLipPuller, AU14_Dimpler, AU15LipCornerDepressor, AU16LowerLipDepressor, AU17ChinRaiser, AU20_LipStretcher, AU23LipTightener, AU24_LipPresser, AU25LipsPart, AU26_JawDrop, AU27_MouthStretch, AU43EyesClosed, PositiveValence, NegativeValence, RotationX, RotationY, RotationZ, TranslationX, TranslationY, TranslationZ

muscle motion with facial shape and appearance information. AUs are basic facial movements that can be visually distinguished and are defined in the facial action coding system [39], [40]. AUs are produced by a single facial muscle or a group of facial muscles [39], [40], [41]. For example, AU12 represents the action of raising the lip corners (as in a smile) and is produced by the facial muscle ‘zygomaticus major’; AU25 represents the mild parting of lips and is produced by either ‘depressor labii inferioris’ or ‘orbicularis oris’; and AU27 represents the stretching or wide opening of the mouth, which is produced by the pterygoids and digastric muscles [39], [40], [41]. Images that show the expressions of AUs are available in [41]. In this study, each facial video in the dataset [7] was analyzed using AUREADER to obtain the 3D head pose and AU intensity estimates for each frame in the video.

IV. METHOD

A. PREPROCESSING, FEATURE EXTRACTION and CLASSICAL MACHINE LEARNING

After the extraction of AUs, 46 channels of information (see Table 2) were available: nEDA, pEDA, HR, BR and eye

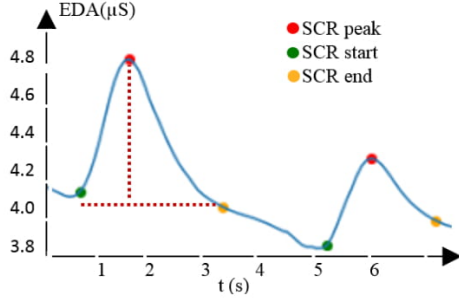


FIGURE 1. Example EDA signal with two skin conductance responses (SCRs). The horizontal (red) dotted line on the first SCR represents the SCR amplitude. The vertical (red) dotted line on the first SCR represents the SCR duration.

tracking data (4 channels), emotional response (8 emotions/channels) and AUREADER data (30 channels). First, all channels were resampled with a sampling frequency of 1 Hz. Next, following the normalization procedure that was used by the dataset creators [1], all channels were normalized via an unsupervised person-specific approach. The normalization function was as follows:

$$Sn_{ij} = \log(S_{ij}) - \log(O_{ij})$$

where O_{ij} represents the overall average value for the i^{th} person and the j^{th} channel, S represents the raw data segments, and Sn represents the normalized data segments. The normalized signals of each driving session were segmented into smaller windows.

Experiments were conducted with windows from 20 seconds up to 80 seconds with a stride of 5 seconds. The segmented data were used as the input to the DL models. For the classical ML models, features were extracted from the segmented data and were used as input to the models.

For each window, the following statistical features were extracted from each channel: the mean, standard deviation, skewness, kurtosis, mean of the first derivative, mean of the second derivative, 25th and 75th percentiles, inter-quartile range, difference between the minimum and the maximum values and coefficient of variation.

Additional features were extracted for the pEDA and the nEDA signals using skin conductance response (SCR) analysis (see Figure 1). This type of feature/analysis is proven to be useful for the detection of stressful conditions in driving scenarios [14] and in practice [42]. The SCR features for each window were the power of the EDA signal, the number of SCRs per second, the power of the SCRs, the sum of the signals' components that have positive derivative, the ratio between the positive derivative and the negative derivative, the mean value of the derivative of the tonic component (the slowly changing EDA component), the mean value of the difference between the raw signal and the tonic component, the total spectral power of the signal in five frequency bands between 0 Hz and 0.6 Hz with a 0.1-Hz span, the amplitude increase of the largest SCR (from the SCR start time to the SCR peak), the amplitude decrease of the largest SCR

peak, the largest SCR increase time, the largest SCR decrease time, the ratio of the increase time and the decrease time of the largest SCR peak, the largest SCR duration, the largest SCR peak increase and decrease slope, the average amplitude increase and decrease of all SCR peaks, and the average amplitude change of all SCRs.

For the classical models, the ML algorithms were used as implemented in the scikit-learn ML toolkit [43]. For each algorithm, parameter tuning was conducted using the following procedure: First, the parameter settings were randomly sampled from distributions that were predefined by an expert. Next, models were constructed with the specified parameters and evaluated using internal k-fold cross-validation on the training data. The search procedure was repeated 10 times. The averaged results are reported in Section V. Experiments were conducted with the following ML algorithms: decision tree [44], RF [45], naïve Bayes [46], KNN [47], SVM [48], bagging [49], adaptive boosting (AdaBoost) [50] and extreme gradient boosting (XGB), which is an updated boosting algorithm. Decision trees were used as the base model for all the ensemble algorithms.

B. DEEP LEARNING

DL represents a class of ML algorithms that use a cascade of multiple layers of nonlinear processing units, which are typically neurons [51]. The first layer receives the input data, and each successive layer accepts the output from the previous layer as input.

The basic strategy dates back to 1943, when McCulloch and Pitts created the first computational model of neural networks (NNs), which was based on threshold logic [52]. Currently, large processing power and memory storage are relatively affordable, and DL models are used to solve complicated artificial intelligence (AI) tasks (e.g., in computer vision, language, biomedicine, and autonomous driving).

1) FULLY CONNECTED NEURAL NETWORKS

A fully connected (FC) NN is a cascade of multiple layers of nonlinear processing units, where each unit receives input from the previous layer. In a typical FC NN, layer i computes an output vector z_i as follows:

$$z_i = f(b_i + W_i z_{i-1}) \quad (1)$$

where b_i (biases) and W_i (weights) are the parameters for the i^{th} layer, z_{i-1} is the output vector of the previous layer and z_0 is the input data. The activation function f can be a rectified linear unit (ReLU) [53]:

$$f(c) = \max(0, c) \quad (2)$$

or another nonlinear function, such as sigmoid or tanh. For classification problems, the final output layer (z_{F_j}) typically uses a softmax activation function.

$$z_{F_j} = \text{softmax}(b_i + W_i z_{i-1}) \quad (3)$$

where j represent the j^{th} row of the weights W_i . The softmax function has the following useful property:

$$\sum_j z_{F_j} = 1 \quad (4)$$

and it is always positive; thus, it can be used as an estimator for the probability that an input pattern x belongs to the j^{th} class for a specified problem:

$$P(Y = j|x) \quad (5)$$

The parameters of the network (b and W) are learned using an optimization algorithm, such as gradient descent [54]. For a binary classification problem, the binary cross-entropy is typically used as a loss function, which is minimized over the pairs of input data/labels (x , y) and predictions p .

$$L = -(y \log(p) + (1 - y) \log(1 - p)) \quad (6)$$

2) CONVOLUTIONAL NEURAL NETWORKS

CNNs are a type of NNs that are designed with three main architectural strategies to ensure various degrees of shift-, scale- and distortion-invariance. This is realized by utilizing (i) local receptive fields, namely, each unit in a layer receives input from a set of neighboring units in the previous layers; (ii) shared weights, namely, units in a layer are organized in groups and all units in the same group share the same set of weights [57], [58]; and (iii) spatial or temporal sampling, namely, if the input is shifted, the feature map output will also be shifted [55]. In addition, due to the specified architecture (parameter sharing and local connections), the CNNs have far fewer connections and parameters to train, while their theoretical best performance is likely to be only slightly worse than that of FC NNs [56].

3) LONG SHORT-TERM MEMORY

Long short-term memory (LSTM) NNs are a type of recurrent neural networks (RNNs), which are networks with memory mechanisms that enable information to persist through time in the model. LSTMs were introduced by Hochreiter and Schmidhuber [67] in 1997. The main processing unit is an LSTM cell, which contains three main gates that regulate the internal cell state and the cell's output. The first gate decides what information should be forgotten (the forget gate) at time t . The decision is made by a sigmoid function, which is applied over the current input x_t and the previous cell output h_{t-1} (Equation 7). The output of the sigmoid function is a number that is between zero and one, where zero corresponds to no propagation.

$$f_t = \sigma(b_f + W_{fx}x_t + W_{fh}h_{t-1}) \quad (7)$$

Next, the input gate (Equation 8) decides what input information will be passed to the output gate via another sigmoid function. The candidate values $\hat{C}_t$ for the new cell state are calculated by a tanh layer (Equation 9). The output of the tanh layer is always between -1 and 1 . The new cell state (C_t) is calculated by multiplying the old state C_{t-1} by f_t to forget some of the previous information and by adding the

element-wise product $i_t * \hat{C}_t$, which consists of the candidate values $\hat{C}_t$, scaled by i_t (Equation 10).

$$i_t = \sigma(b_i + W_{ix}x_t + W_{ih}h_{t-1}) \quad (8)$$

$$\hat{C}_t = \tanh(b_c + W_{cx}x_t + W_{ch}h_{t-1}) \quad (9)$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (10)$$

Finally, the output gate decides which parts of the cell state (C_t) it is going to output (propagate) via another sigmoid layer (Equation 11), and the final output of the cell (Equation 12) is calculated by applying tanh on the current cell state and scaling it with o_t from Equation (11).

$$o_t = \sigma(b_o + W_{ox}x_t + W_{oh}h_{t-1}) \quad (11)$$

$$h_t = o_t * \tanh(C_t) \quad (12)$$

The equations above represent the main strategy of LSTMs. Additionally, there are many variations of RNNs and LSTMs [68], [69], [70].

4) DEEP LEARNING ARCHITECTURES

DL realized a breakthrough performance at solving pattern recognition problems [59], especially in image [56], [60], [63] and natural language processing (NLP) [61], [62]. For example, DL was used to realize image super resolution [64]. In another study, DL was used for "seeing in darkness", which is a technique for reconstructing and brightening dark images [65]. For NLP, Google introduced BERT – a state-of-the-art method for "language understanding" [66]. However, DL architectures for signal processing have not yet realized such a breakthrough and designing them remains challenging, especially for problems with limited data. The layered structure of the NNs enables the construction a variety of DL architectures by combining layers. For example, ConvLSTM stacks CNN layers on top of LSTM layers, namely, the input is received by the CNN layers and propagated to the LSTM layers. In addition to the vertical stacking, one can also experiment with horizontal stacking. For example, for a 2-channel dataset, one can use a ConvLSTM for each channel and later fuse the outputs of the two ConvLSTMs using an FC layer. Which DL architecture is most suitable depends on the dataset; thus, extensive experimentation is required.

Figure 2 presents the two fusion approaches that are evaluated in the experiments. The early-fusion approach merges all 46 channels at the input regardless of the modality. Then, the merged input data are fed into DL layers. The DL layers can be FC layers, CNN layers or LSTM layers. The mid-fusion approach uses DL layers that are specific for each modality, and later, the modality-specific layers are fused using a general DL. The early-fusion approach learns shared weights for all input modalities, whereas the mid-fusion approach initially learns separate weights for each modality (represented by purple squares in Figure 2) and later learns shared weights (represented by orange squares in Figure 2). An additional DL architecture that is evaluated in the experiments is the spectro-temporal ResNet (STRNet), which is an architecture that was successfully applied on sensor data

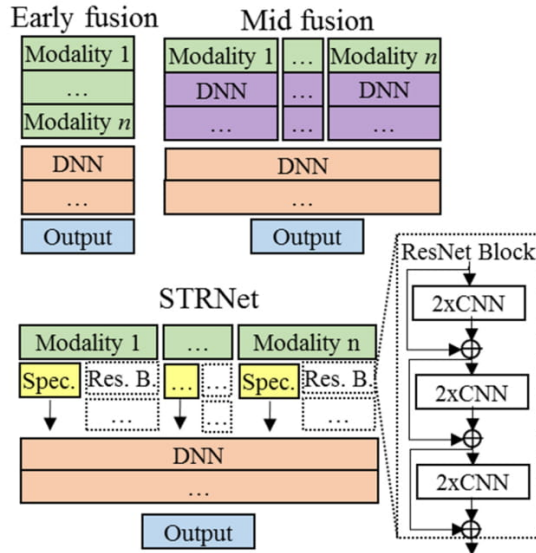


FIGURE 2. Two types of DL fusion approaches that are used in the study: early fusion and mid-fusion. The spectro-temporal ResNet (STRNet) is a special case of the mid-fusion approach.

TABLE 3. Deep learning architectures that are used in the study.

Classifier	Architecture
eCNN	2 x CNN(128) - FC(64)
eLSTM	2 x LSTM(128) - FC(64)
eConvLSTM	2 x CNN(128) - LSTM(128) - FC(64)
mCNN	N x 2 x CNN(128) - FC(64)
mLSTM	N x 2 x LSTM(128) - FC(64)
mConvLSTM	N x 2 x CNN(128) - LSTM(128) - FC(64)
STRNet	N x [3 x (2 x CNN(64))] - FC(64)

in previous study on human activity recognition from smart-phone sensors [71], for chronic heart failure detection from heart sounds [72], and for blood pressure estimation from photoplethysmogram (PPG) data [73].

STRNet is a special type of mid-fusion network in which each modality is associated with two branches: one that evaluates the raw sensor signal in the time domain using residual blocks [74] and another that evaluates a spectral representation of the signal. Toward the end of the network, the two branches, namely, the spectral and the temporal branches of each modality, are merged using FC layers.

The structures of the DL architectures that are used in this study are presented in Table 3. There are three early-fusion architectures (eCNN, eLSTM and eConvLSTM) and four mid-fusion architectures (mCNN, mLSTM and mConvLSTM and STRNet). For example, the architecture “2 x CNN(128) - FC(64)” contains two CNN layers, each with 128 filters, and one FC layer with 64 neurons. N represents the number of input channels. All DL architectures

contain batch normalization layers [75] to reduce the internal covariance shift, ReLU activation layers [53] to accelerate the training process; maximum pooling layers for dimensionality reduction, and a final softmax layer, which outputs the estimated class probability for distracted vs. not distracted driving. The DL architectures are available online <https://repo.ijs.si/martingjoreski/driving-distractions/tree/master/DL%20architectures>. All DL models were trained by minimizing the binary cross-entropy loss function using the Adam optimizer with a learning rate of 10^{-5} and a decay of 10^{-3} . The batch size was set to 256 with a maximum number of training epochs of 30.

V. EXPERIMENTS

First, a statistical analysis of the input signals was conducted to analyze the relations between the modalities and the driving distractions. Next, ML analysis was conducted to compare classical ML and DL for the detection of driving distractions. Next, ML analysis was conducted to compare classical ML and DL for the detection of driving distractions.

For the ML analysis, the data of the first 10 subjects were used as the test set (close to 20% of the overall data), and the data of the remaining subjects were used as the training set. Thus, the classifiers are subject-independent. Each ML algorithm was evaluated in the construction of two types of classifiers:

- A window classifier: Outputs a prediction whether distraction was detected for each input window (binary classification). This classifier would be useful for monitoring driver distractions in real time;
- A session classifier: Outputs only one prediction per driving session, namely, each driving session is classified as ‘with distractions’ or ‘without distractions’. The decision is based on the predictions of the window classifier that are obtained using a threshold logic. The thresholds were optimized for each classifier using cross-validation on the training set. This classifier would be useful for the offline determination of whether there was a distraction present during the past driving session.

The ground-truth labels were determined using the following rules: (i) the instances for the window classifiers are labeled as positive, namely, distracted driving should be detected, if a distraction was present in at least 5 seconds of the input window and (ii) the instances for the session classifiers are labeled as positive if a distraction was present for at least 5 seconds of the overall driving session. For the window classifiers, one instance is one window (segment) that was extracted using an overlapping sliding window with a 5-second stride; thus, a prediction is output every 5 seconds.

For the session classifiers, one instance is one session. For example, Table 4 summarizes the experimental data (instances) that are produced after using an overlapping sliding window of 60 seconds with a stride of 5 seconds.

Experiments were conducted with window sizes from 20 to 80 seconds. F1-score was used to evaluate the classifiers

TABLE 4. Experimental data. The sizes of the training and test subsets for the window classifiers and for the session classifiers.

Instances	Subset	Distraction		Majority
		No	Yes	
Window	Test	3724	2135	64 (%)
	Training	13144	8023	62 (%)
Session	Test	21	34	62 (%)
	Training	66	128	66 (%)

(Equations 13 to 15):

$$\text{Precision} = 100 * \frac{\text{Correctly detected distrction instances}}{\text{Overall predicted distractions}} \quad (13)$$

$$\text{Recall} = 100 * \frac{\text{Correctly detected distrction instances}}{\text{Overall distraction instances}} \quad (14)$$

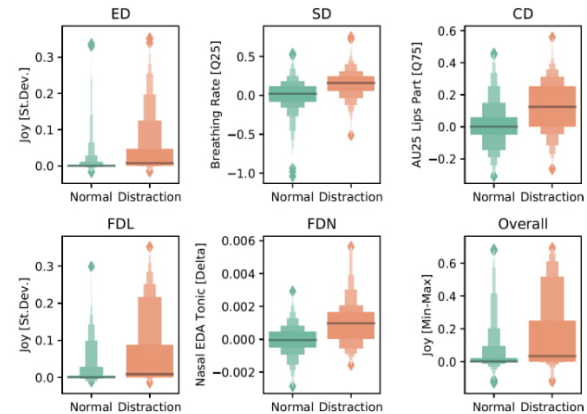
$$F1 - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (15)$$

A. INPUT ANALYSIS

In the initial dataset study [1], the authors showed that there is a statistically significant difference in the mean values of the nEDA when measured in the normal segments of the driving sessions, compared to the distracted segments of the same driving sessions. Inspired by that analysis, statistical tests were conducted in this study to determine whether such a statistically significant difference is present for the remaining features in the experiments. For the statistical analysis, the Wilcoxon signed-rank test was used, which is an alternative to the paired Student's t-test that lacks the t-test's normality assumption on the distribution of the paired differences. The Wilcoxon test is a non-parametric statistical hypothesis test that is used to determine whether two paired samples are sampled from the same distribution [76]. In this experimental setting, one sample contains values of a specified feature that was extracted from the normal segments of the driving sessions, and the other sample contains values for the same feature that were extracted from the distraction segments of the same driving session. Informative features should differ in terms of their distributions when conditioned on the type of the segment (with vs. without distraction). The tests showed for 177 of the 562 features, the test p-value was smaller than 0.001; these are named "informative features". Table 5 presents the top three modalities for each type of driving session (ED, SD, CD, FDL and FDN) and for all driving sessions (Overall). The modalities are ranked using the ratio of informative features per modality. According to the table, nEDA is ranked among the top 3 for each driving session. The emotions are ranked among the top 3 for five out of the six types of driving sessions. The facial action units (AUs) are ranked among the top 3 for three types of driving sessions (ED, CD and FDN). For recognizing the mixed distractions in the failure sessions (FDL and FDN), nEDA is the most informative modality. Overall, when all driving sessions are joined

TABLE 5. Top three modalities for each type of driving session and overall) ranked according to the ratio of informative features.

Session	Modality	Informative features (%)
ED	Emotion	22/87 (25.3)
	nEDA	9/39 (23.1)
	Face AUs	27/305 (1.8)
SD	Breathing	4/11 (36.4)
	Eye Tr.	8/32 (25.0)
	nEDA	6/39 (15.4)
CD	Emotion	22/87 (25.3)
	nEDA	5/39 (12.8)
	Face AUs	29/305 (9.5)
FDL	nEDA	4/39 (10.3)
	Breathing	1/11 (9.1)
	Emotion	2/87 (2.3)
FDN	nEDA	3/39 (7.7)
	Emotion	2/87 (5.1)
	Face AUs	3/305 (1.0)
Overall	Emotion	47/87 (54)
	nEDA	18/39 (46.2)
	Face AUs	47/305 (15.4)

**FIGURE 3.** Distributions of the most informative features, namely, the features with the smallest p-value, for each type of driving session (ED, SD, CD, FDL, FDN) and overall.

and the statistical tests are conducted for normal segments vs. distraction segments, the two most informative modalities are the recognized emotions and nEDA. This is followed by the facial AUs in the third position.

Figure 3 presents the distributions of the most informative features, namely, the features with the smallest p-value for each type of driving session (ED, SD, CD, FDL, FDN) and for all driving sessions (Overall).

The distributions are represented as boxenplots (letter-value-plots), which provide a better representation of the distribution of the data than boxplots when outlier values are present [77]. According to the figure, for recognizing an emotional distraction (ED), the most informative feature is the standard deviation of the activation of the emotion "joy".

TABLE 6. Evaluation results for the classical ML classifiers and DL models. F1-scores of the window classifiers (F1) and F1-score of the session classifiers (F1-s).

Type	Classifier	F1 (%)	F1-s (%)
classical ML	RF	67	84
classical ML	GB	73	87
classical ML	KNN	50	78
classical ML	NB	63	74
classical ML	DT	56	66
classical ML	Bagging	64	82
classical ML	XGB	72	88
end-to-end DL	eCNN	65	68
end-to-end DL	eLSTM	67	75
end-to-end DL	eConvLSTM	65	69
end-to-end DL	mCNN	64	69
end-to-end DL	mLSTM	64	70
end-to-end DL	mConvLSTM	62	69
end-to-end DL	STRNet	67	80

Thus, during the distraction segments, the subjects showed an increased standard deviation of this emotion. Second, for recognizing a sensorimotor distraction (SD), the most informative feature is the 25th percentile of the subjects' BR. During the distraction segments, the subjects had an increased BR. Third, for recognizing a cognitive distraction (CD), the most informative feature is the 75th percentile of intensities of AU25 "Lips Part". During the distraction segments, the subjects showed increased lip movement. This could be because the cognitive-distraction sessions involved speech, which – if true – may be regarded as an artifact of the dataset rather than a general finding. Fourth, for recognizing the mixed distractions in failure session FDL, the most informative feature is the standard deviation of the activation of the emotion "joy", which is the same as for the ED.

Fifth, for recognizing the brake failure in session FDN, the most informative feature is the first derivative of the tonic component of nEDA. An increased positive derivative corresponds to more sweating of the subjects during the brake failure. Finally, for recognizing general distractions, the most informative feature is the difference between the minimum and the maximum values of the activation of the emotion "joy". This may indicate that the subjects had stronger emotional responses during the distraction segments.

B. MACHINE-LEARNING ANALYSIS

In the initial experiments, seven classical ML algorithms and seven end-to-end DL algorithms were compared for the detection of driving distraction (binary classification). The eye tracking data were not used in these experiments because the data were missing for more than 50% of the sessions. An overlapping sliding window of 20 seconds with a 5-second stride was used in these experiments. The results are presented in Table 6. Column F1 presents the F1-scores

that are realized by the window classifiers, and column F1-s presents the F1-scores that are realized by the session classifiers. According to the results, the highest scores are realized by the classical ML classifiers, namely, GB and XGB. The highest F1-score for the window classifiers is 73%, and the highest F1-score for the session classifiers (column F1-s) is 88%. Among the DL classifiers, eLSTM and STRNet have similar performance, with an F1-score of 67% realized by the window classifiers.

The eLSTM session classifier realized an F1-score of 75%, and the STRNet session classifier realized an F1-score of 80%. Compared to the classical classifiers, eLSTM and STRNet outperformed the KNN, NB, DT and Bagging classifiers and were outperformed RF, GB and XGB. The experiments did not show a clear preference for the use of early or mid-fusion by the DL classifiers (denoted by the prefixes 'e' and 'm' in Table 3).

Next, a more detailed evaluation was conducted for the two best-performing classical classifiers and the two best-performing DL classifiers. Tests were conducted with various window sizes and input signals (modalities). The results are presented in Table 7. The first column presents the size of the input temporal segment in seconds (varied from 20 seconds to 80 seconds), the second column presents the ML algorithm, and the remaining columns present the F1-scores of the window classifiers (F1) and the F1-scores of the session classifiers (F1-s) for each of the input categories: face AUs, emotional activation (EMO), heart rate (HR), breathing (BR), nEDA and pEDA. For the column "All", all features/modalities were used as input to the classifiers. For the column "Selected", only the statistically significant features/modalities were used as input. According to Table 7, no classifiers perform well when only one of the physiological signals (EDA, nEDA and BR) is used as input, except the session HR classifier. The classical classifiers outperform the DL classifiers overall. Regarding the window classifiers, the highest F1 score of 79% is realized by the two classical classifiers, namely, XGB and GB, using the AUs as an input with a window size of 60 seconds. Regarding the session classifiers, the highest F1-score (F1-s) of 94% is realized by XGB using the AUs as an input with a window size of 60 seconds. Hence, the visual modalities are the most informative modalities in the experimental dataset. Among the DL classifiers, the highest performance is realized by STRNet using the selected signals and a window size of 60 seconds. The F1-score of the window classifier is 75%, and the F1-score of the session classifier (F1-s) is 87%.

Regarding the size of the input windows, all classifiers perform better with longer windows (40 seconds to 80 seconds), which is probably because longer windows contain more information.

This is especially true for the DL classifiers. Figure 4 presents the precision-recall curves of the best-performing classifiers, namely, the window classifier and the session classifier that were built with XGB using AUs as input with a window size of 60 seconds. Such curves would be useful

TABLE 7. Evaluation results for two best-performing classical classifiers and the two best-performing DL classifiers. The first column presents the size of the temporal segment in seconds, the second column presents the ML algorithm and the remaining columns present the F1-scores (%) of the window classifiers (F1) and the F1-scores (%) of the session classifiers (F1-s) for each of the specified inputs.

Win.	Classifier	All		Selected		AUs		EMO		HR		BR		nEDA		pEDA	
		F1	F1-s	F1	F1-s	F1	F1-s	F1	F1-s	F1	F1-s	F1	F1-s	F1	F1-s	F1	F1-s
20	GB	73	87	73	87	76	92	73	87	61	76	55	54	63	66	42	27
	XGB	72	88	72	88	76	90	72	82	61	76	55	65	63	63	42	27
	eLSTM	67	75	64	65	53	70	62	69	42	28	58	67	52	46	42	28
	STRNet	67	80	68	83	51	65	65	79	48	41	60	74	52	48	42	28
40	GB	77	85	77	85	79	92	75	88	64	78	57	55	65	65	41	30
	XGB	75	88	75	88	79	88	74	92	63	74	57	60	65	65	41	26
	eLSTM	70	74	68	71	54	67	66	72	55	52	59	54	63	60	42	28
	STRNet	72	82	73	87	53	61	66	77	48	41	60	74	63	64	42	28
60	GB	77	88	77	88	78	86	74	86	65	79	59	58	65	63	45	33
	XGB	76	88	76	88	78	94	74	83	66	83	59	54	65	65	45	38
	eLSTM	69	72	64	67	55	63	63	66	50	51	66	78	62	64	45	38
	STRNet	74	86	75	87	54	65	66	72	58	57	67	77	61	63	42	28
80	GB	75	92	75	92	78	92	71	82	69	83	61	64	63	65	49	33
	XGB	77	94	77	94	78	90	62	69	68	83	62	69	65	65	49	33
	eLSTM	66	71	64	67	55	61	64	70	55	52	62	69	58	59	42	33
	STRNet	72	82	73	85	56	56	66	71	58	57	69	74	60	61	42	28

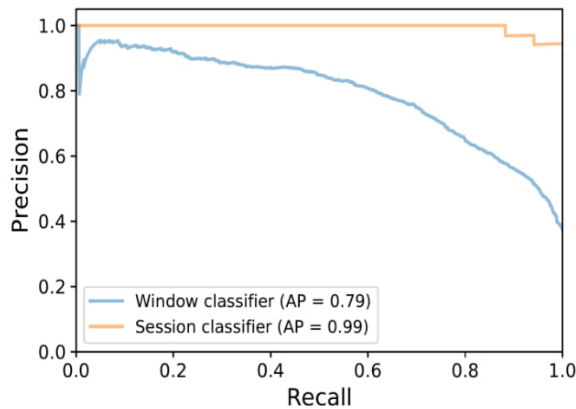


FIGURE 4. Precision-recall curves of the best-performing window classifier (blue) and session classifier (orange) that are built with XGB. AP denotes the average precision, which is defined as $\sum_n (R_n - R_{n-1}) / P_n$, where R_n and P_n are the recall and precision, respectively, for the n^{th} decision threshold.

for the modification of the decision threshold. For example, in various cases, higher recall might be preferred over lower precision, as undetected distractions (false negatives) might be more dangerous than falsely detected distractions (false positives).

VI. DISCUSSION

The best-performing classical ML classifiers outperformed the best-performing DL classifiers. There may be two main reasons for this: (i) The size of the dataset is not sufficient for the end-to-end learning to outperform the best-performing classical ML classifiers. According to Table 3, the models

were trained on close to 20,000 instances. While this is a large number of instances compared to related affective computing studies, which typically use a few thousand instances, it is 750 times smaller than ImageNet, which is the dataset that is used to train state-of-the-art DL NNs for image processing. (ii) DL excels in pattern recognition (e.g., image classification, object detection, and face recognition). In this use case, “pattern recognition”, namely, emotion recognition and facial AU extraction, was conducted with other modules, and the extracted information was fed into both the classical and the DL classifiers. The access to this information probably gave the classical ML an edge as it can learn better from smaller datasets. The STRNet consistently outperformed all other classifiers when using the breathing rate (BR) as input. This is likely because spectral-domain information is especially important in relation to BR, and STRNet is the only classifier that uses time- and spectral-domain information. Classical ML models use only statistical features (except the pEDA and nEDA features), and the other DL architectures use only signals in the time domain.

The feature selection can significantly influence the classification performance of the classical feature-based ML methods. In this study, ranking-based feature selection (also known as filter methods) was used, as it is computationally efficient and does not require a classifier for feature selection. In contrast, the filter methods estimate the quality of each feature separately; hence, they fail to consider useful feature combinations. This may be the reason why the classical ML models that were built with pre-selected features did not realize the best performance. Wrapper-based feature selection methods or combinations of filter- and wrapper-based methods [71] may be useful in this case.

TABLE 8. Percentages of correctly classified instances by the best-performing window classifier and session classifier. the last row presents the accuracies of the classifiers.

Session Type	Classifier	
	Window (%)	Session (%)
ED	72	100
SD	80	100
CD	67	89
FDL	73	100
FDN	69	100
RD	93	90
ND	90	90
Overall (Accuracy)	80	95

Table 8 presents the percentages of correctly classified instances by the best-performing classifiers. The window classifier correctly classifies the windows from the normal driving sessions (ND and RD) with an average percentage of 92%, which is significantly higher than the average percentage of correctly classified windows from the distracted sessions (ED, SD, CD, FDL and FDN), which is 72%. This is probably due to the noise in the labels that is present in the windows from the distracted sessions. All windows from the normal driving sessions have the label “normal”. However, to derive the labels of the windows from the distracted driving sessions, the following rule was used: if a distraction was present for at least 5 seconds of the window, the window should be classified as distracted, and it should be classified as normal otherwise. In various cases, the subject may need more than 5 seconds for the distraction to induce an affective response. Thus, due to the absence of an affective response, the normal windows are same as the distracted windows when analyzed using the physiological and the visual sensors. To mitigate this problem, one might use methods that explicitly incorporate label jitter into the model training process [78]. The label jitter may be why the performance on the physiological signals is worse than that on the visual signals for the detection of the driving distractions. The physiological signals may have a longer latency, namely, it may take longer for the distraction (stressor) to induce a change in the physiological signals than in the visual signals.

The session classifiers outperform the window classifiers mostly because the window classifiers must detect the exact time when the distraction occurred. In contrast, the timing is not important for the session classifiers; they need to detect only some of the distractions, which also mitigates the label-jitter problem.

For recognizing the cognitive distraction, the most informative feature was related to the driver’s increased lip movement. This is expected since the cognitive-distraction sessions involved answering questions. One should be careful with using only this feature for the detection of distraction segments, as this finding may be regarded as scenario overfitting rather than a general finding. Another interesting finding is

TABLE 9. Abbreviations.

Action Units	AUs
Adaptive Boosting	AdaBoost
Artificial Intelligence	AI
Breathing Rate	BR
Cognitive Drive	CD
Convolutional Neural Networks	CNNs
Deep Learning	DL
Electrocardiogram	ECG
Electrodermal Activity	EDA
Electroencephalogram	EEG
Electrooculogram	EOG
Emotional Activation	EMO
Emotional Drive	ED
Extreme Gradient Boosting	XGB
F1-Scores Of The Session Classifiers	F1-s
Failure-Loaded Drive	FDL
Failure-Normal Drive	FDN
Fully Connected	FC
Heart Rate	HR
Long Short-Term Memory Neural Networks	LSTMs
Machine Learning	ML
Naïve Bayes	NB
Nasal Electrodermal Activity	nEDA
Neural Network	NN
Normal Drive	ND
Palm Electrodermal Activity	pEDA
Principal Component Analysis	PCA
Relaxed Drive	RD
Sensorimotor Drive	SD
Skin Conductance Response	SCR
Spectro-Temporal Residual Network	STRNet
Support Vector Machine	SVM

that for recognizing general distractions, the most informative feature is related to the activation of the emotion “joy”. A more detailed analysis showed that this emotion had both higher average values and a higher standard deviation for the distracted (stressful) segments than for the normal segments. This may indicate that the normal driving sessions were more boring for the participants, whereas the driving sessions that contained distractions were more fun; as this was a driving simulation study, no distraction was regarded as dangerous by the subjects. These findings raise more general concerns regarding the generalization performances of systems that have been trained on a single dataset that was collected in a single environment. Such systems may classify any type

of motion, speech, or emotional activation as a “distraction” because they were trained only for distraction detection.

In this study, all inputs were represented as 1D signals; thus, specialized DL architectures for time series were used. In the future, comparison of the methods should be made with DL classifiers that detect driver distractions directly from images. Additionally, generative adversarial networks (GANs) and transfer learning may be used to improve the performance of the DL classifiers. Furthermore, since the best-performing classifier in this study was built using AUs, in the future, different fusion strategies can be tested for the extraction of higher-level semantic facial activities (e.g., speaking, listening, and concentrating) with more semantic content.

VII. CONCLUSION

This paper presented an analysis for the determination of which ML methods perform best in detecting various driving distractions using which sensors and which data-capture methods, with a focus on physiological sensors and sensors that are based on video cameras. The statistical analysis showed that the most informative feature/modality for detecting driver distraction depends on the type of distraction. Overall, the video-based modalities were most informative, and classical ML classifiers realized high performance using one of the video-based modalities. In contrast, the DL classifiers require more modalities, namely, either all modalities or pre-selected modalities, for the construction of useful classifiers. For the analyzed data, the classical ML (XGB using the AUs as an input with a window size of 60 seconds) realized high performance and outperformed DL methods; hence, the detection of driver distractions may be technically feasible with the current knowledge. A demo of the final ML classifier is available online.² Finally, problems such as label jitter, scenario overfitting and unsatisfactory generalization performance were identified and discussed to provide guidance for future studies in this area.

APPENDIX

See Table 9.

REFERENCES

- [1] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, Standard J3016, SAE International, 2018.
- [2] C. Peter and R. Beale, *Affect and Emotion in Human-Computer Interaction: From Theory to Applications*. Berlin, Germany: Springer, 2008.
- [3] F. Eyben, M. Wöllmer, T. Poitschke, B. Schuller, C. Blaschke, B. Färber, and N. Nguyen-Thien, “Emotion on the road—Necessity, acceptance, and feasibility of affective computing in the car,” *Adv. Hum.-Comput. Interact.*, vol. 2010, pp. 1–17, Jul. 2010.
- [4] M. A. Regan, C. Hallett, and C. P. Gordon, “Driver distraction and driver inattention: Definition, relationship and taxonomy,” *Accident Anal. Prevention*, vol. 43, no. 5, pp. 1771–1781, Sep. 2011.
- [5] R. J. Hanowski, R. L. Olson, J. S. Hickman, and J. Bocanegra, “Driver distraction in commercial motor vehicle operations,” in *Driver Distraction and Inattention*. Boca Raton, FL, USA: CRC Press, 2017, pp. 141–156.
- [6] S. Masood, A. Rai, A. Aggarwal, M. N. Doja, and M. Ahmad, “Detecting distraction of drivers using convolutional neural network,” *Pattern Recognit. Lett.*, early access, Jan. 12, 2018, doi: 10.1016/j.patrec.2017.12.023.
- [7] I. Pavlidis, M. Deosta, S. Taamneh, M. Manser, T. Ferris, R. Wunderlich, E. Akleman, and P. Tsiamyrtzis, “Dissecting driver behaviors under cognitive, emotional, sensorimotor, and mixed stressors,” *Sci. Rep.*, vol. 6, no. 1, May 2016, Art. no. 25651.
- [8] J. Gomez, D. Akleman, E. Akleman, and I. Pavlidis, “Causality effects of interventions and stressors on driving behaviors under typical conditions,” *Mathematics*, vol. 6, no. 8, p. 139, 2018.
- [9] A. Fernández, R. Usamentiaga, J. Carús, and R. Casado, “Driver distraction using visual-based sensors and algorithms,” *Sensors*, vol. 16, no. 11, p. 1805, 2016.
- [10] B. Goel, A. Dey, P. Bharti, K. Ahmed, and S. Chellappan, “Detecting distracted driving using a wrist-worn wearable,” in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2018, pp. 233–238.
- [11] B.-G. Lee and W.-Y. Chung, “Driver alertness monitoring using fusion of facial features and bio-signals,” *IEEE Sensors J.*, vol. 12, no. 7, pp. 2416–2422, Jul. 2012, doi: 10.1109/JSEN.2012.2190505.
- [12] C.-T. Lin, Y.-C. Chen, T.-Y. Huang, T.-T. Chiu, L.-W. Ko, S.-F. Liang, H.-Y. Hsieh, S.-H. Hsu, and J.-R. Duann, “Development of wireless brain computer interface with embedded multitask scheduling and its application on real-time driver’s drowsiness detection and warning,” *IEEE Trans. Biomed. Eng.*, vol. 55, no. 5, pp. 1582–1591, May 2008, doi: 10.1109/TBME.2008.918566.
- [13] R. N. Khushaba, S. Kodagoda, S. Lal, and G. Dissanayake, “Driver drowsiness classification using fuzzy wavelet-packet-based feature-extraction algorithm,” *IEEE Trans. Biomed. Eng.*, vol. 58, no. 1, pp. 121–131, Jan. 2011, doi: 10.1109/TBME.2010.2077291.
- [14] J. A. Healey and R. W. Picard, “Detecting stress during real-world driving tasks using physiological sensors,” *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 2, pp. 156–166, Jun. 2005, doi: 10.1109/TITS.2005.848368.
- [15] R. K. Singh, A. Sarkar, and C. S. Anoop, “A health monitoring system using multiple non-contact ECG sensors for automotive drivers,” in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2016, pp. 1–6, doi: 10.1109/I2MTC.2016.7520539.
- [16] B. G. Lee, J.-H. Park, C. C. Pu, and W.-Y. Chung, “Smartwatch-based driver vigilance indicator with kernel-fuzzy-C-means-wavelet method,” *IEEE Sensors J.*, vol. 16, no. 1, pp. 242–253, Jan. 2016, doi: 10.1109/JSEN.2015.2475638.
- [17] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. E. Lopez, “Real-time system for monitoring driver vigilance,” *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 63–77, Mar. 2006, doi: 10.1109/TITS.2006.869598.
- [18] M. Omidyeganeh, S. Shirmohammadi, S. Abtahi, A. Khurshid, M. Farhan, J. Scharcanski, B. Hariri, D. Laroche, and L. Martel, “Yawning detection using embedded smart cameras,” *IEEE Trans. Instrum. Meas.*, vol. 65, no. 3, pp. 570–582, Mar. 2016, doi: 10.1109/TIM.2015.2507378.
- [19] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, “Driver gaze tracking and eyes off the road detection system,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2014–2027, Aug. 2015, doi: 10.1109/TITS.2015.2396031.
- [20] E. Murphy-Chutorian and M. M. Trivedi, “Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 300–311, Jun. 2010, doi: 10.1109/TITS.2010.2044241.
- [21] P. Smith, M. Shah, and N. da Vitoria Lobo, “Determining driver visual attention with one camera,” *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 4, pp. 205–218, Dec. 2003, doi: 10.1109/TITS.2003.821342.
- [22] C. Yan, F. Coenen, and B. Zhang, “Driving posture recognition by convolutional neural networks,” *IET Comput. Vis.*, vol. 10, no. 2, pp. 103–114, Mar. 2016, doi: 10.1049/iet-cvi.2015.0175.
- [23] B.-L. Lee, B.-G. Lee, and W.-Y. Chung, “Standalone wearable driver drowsiness detection system in a smartwatch,” *IEEE Sensors J.*, vol. 16, no. 13, pp. 5444–5451, Jul. 2016, doi: 10.1109/JSEN.2016.2566667.
- [24] A. Aksjonov, P. Nedoma, V. Vodovozov, E. Petlenkov, and M. Herrmann, “Detection and evaluation of driver distraction using machine learning and fuzzy logic,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2048–2059, Jun. 2019.
- [25] Y. Saito, M. Itoh, and T. Inagaki, “Driver assistance system with a dual control scheme: Effectiveness of identifying driver drowsiness and preventing lane departure accidents,” *IEEE Trans. Human-Machine Syst.*, vol. 46, no. 5, pp. 660–671, Oct. 2016, doi: 10.1109/THMS.2016.2549032.

²<https://repo.ijs.si/martingjoreski/driving-distractions/blob/master/Demo.mp4>

- [26] N. Apostoloff and A. Zelinsky, "Vision in and out of vehicles: Integrated driver and road scene monitoring," *Int. J. Robot. Res.*, vol. 23, nos. 4–5, pp. 513–538, 2004.
- [27] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Driver behavior profiling using smartphones: A low-cost platform for driver monitoring," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 1, pp. 91–102, Apr. 2015, doi: 10.1109/ITS.2014.2328673.
- [28] J. Wang, C. Yu, S. E. Li, and L. Wang, "A forward collision warning algorithm with adaptation to driver behaviors," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1157–1167, Apr. 2016, doi: 10.1109/ITS.2015.2499838.
- [29] G. Sikander and S. Anwar, "Driver fatigue detection systems: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2339–2352, Jun. 2019.
- [30] A. Borbély, "A two process model of sleep regulation," *Hum. Neurobiol.*, vol. 1, no. 3, pp. 195–204, 1982.
- [31] A. J. Belyavin and M. B. Spencer, "Modeling performance and alertness: The QinetiQ approach," *Aviation, Space Environ. Med.*, vol. 75, no. 3, pp. A93–A103, 2004.
- [32] T. Azim, M. A. Jaffar, and A. M. Mirza, "Fully automated real time fatigue detection of drivers through fuzzy expert systems," *Appl. Soft Comput.*, vol. 18, pp. 25–38, May 2014.
- [33] A. Koesdwiady, S. M. Bedawi, C. Ou, and F. Karray, "End-to-end deep learning for driver distraction recognition," in *Proc. Int. Conf. Image Anal. Recognit.* Cham, Switzerland: Springer, 2017, pp. 11–18.
- [34] M. S. Majidi, S. Ram, J. T. Gill, and J. J. Rodriguez, "Drive-net: Convolutional network for driver distraction detection," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation (SSIAI)*, Apr. 2018, pp. 1–4.
- [35] M. Hsayeni, S. Saxena, R. Ptucha, and A. Savakis, "Distracted driver detection: Deep learning vs handcrafted features," *Electron. Imag.*, vol. 2017, no. 10, pp. 20–26, Jan. 2017.
- [36] Y. Zhou, P. Tsiamyrtzis, P. Lindner, I. Timofeyev, and I. Pavlidis, "Spatiotemporal smoothing as a basis for facial tissue tracking in thermal imaging," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 5, pp. 1280–1289, May 2013.
- [37] D. Shastri, M. Papadakis, P. Tsiamyrtzis, B. Bass, and I. Pavlidis, "Perinatal imaging of physiological stress and its affective potential," *IEEE Trans. Affect. Comput.*, vol. 3, no. 3, pp. 366–378, Jul. 2012.
- [38] T. Hassan, D. Seuss, J. Wollenberg, J. Garbas, and U. Schmid, "A practical approach to fuse shape and appearance information in a Gaussian facial action estimation framework," in *Proc. 22nd Eur. Conf. Artif. Intell., 9th Int. Conf. Prestigious Appl. Artif. Intell. (PAIS)*, vol. 285, G. A. Kaminka, Ed., 2016, pp. 1812–1817.
- [39] P. Ekman and W. V. Friesen, *Facial Action Coding System*. Palo Alto, CA, USA: Consulting Psychologists Press, 1978.
- [40] P. Ekman, W. V. Friesen, and J. C. Hager, *The Facial Action Coding System*, 2nd ed. Salt Lake City, UT, USA: Research Nexus eBook, 2002.
- [41] J. F. Cohn, Z. Ambadar, and P. Ekman, "Observer-based measurement of facial expression with the facial action coding system," in *Handbook of Emotion Elicitation and Assessment* (Series in Affective Science), J. A. Coan and J. J. B. Allen, Eds. London, U.K.: Oxford Univ. Press, 2007, pp. 203–221.
- [42] M. Gjoreski, M. Luštrek, M. Gams, and H. Gjoreski, "Monitoring stress with a wrist device using context," *J. Biomed. Informat.*, vol. 73, pp. 159–170, Sep. 2017.
- [43] *Scikit-Learn*. [Online]. Available: <https://scikit-learn.org/stable/>
- [44] J. R. Quinlan, "Improved use of continuous attributes in C4.5," *J. Artif. Intell. Res.*, vol. 4, pp. 77–90, Mar. 1996.
- [45] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1, Aug. 1995, pp. 278–282.
- [46] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Kuala Lumpur, Malaysia: Pearson Education Limited, 2016.
- [47] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991.
- [48] J. Shawe-Taylor and N. Cristianini, *An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods*, vol. 204. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [49] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
- [50] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proc. Eur. Conf. Comput. Learn. Theory*. Berlin, Germany: Springer, 1995, pp. 23–37.
- [51] L. Deng, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, 2014, doi: 10.1561/20000000039.
- [52] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.
- [53] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 1–8.
- [54] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.
- [55] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [57] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 609–616.
- [58] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2146–2153.
- [59] X. Jiang, Y. Pang, X. Li, and J. Pan, "Speed up deep neural network based pedestrian detection by sharing features across multi-scale models," *Neurocomputing*, vol. 185, pp. 163–170, Apr. 2016.
- [60] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.
- [61] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.
- [62] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.
- [63] S. Chaib, H. Yao, Y. Gu, and M. Amrani, "Deep feature extraction and combination for remote sensing image classification based on pre-trained CNN models," in *Proc. 9th Int. Conf. Digit. Image Process. (ICDIP)*, Jul. 2017, Art. no. 104203.
- [64] T. Vandal, E. Kodra, S. Ganguly, A. Michaelis, R. Nemani, and A. R. Ganguly, "DeepSD: Generating high resolution climate change projections through single image super-resolution," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1663–1672.
- [65] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3291–3300.
- [66] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [67] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [68] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 3, Jul. 2000, pp. 189–194.
- [69] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, doi: 10.1162/089976600300015015.
- [70] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "Learning phrase representations using RNN encoder-decoder," in *Proc. Conf. Empirical Methods Natural Lang. Process. Stat. Mach. Transl.*, 2014, pp. 1724–1734.
- [71] M. Gjoreski, V. Janko, G. Slapnicar, M. Mlakar, N. Rescic, J. Bizjak, V. Drobnic, M. Marinko, N. Mlakar, M. Luštrek, and M. Gams, "Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors," *Inf. Fusion*, 2020.
- [72] M. Gjoreski, A. Gradisek, B. Budna, M. Gams, and G. Poglajen, "Machine learning and end-to-end deep learning for the detection of chronic heart failure from heart sounds," *IEEE Access*, vol. 8, pp. 20313–20324, 2020, doi: 10.1109/ACCESS.2020.2968900.
- [73] G. Slapnicar, N. Mlakar, and M. Luštrek, "Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network," *Sensors*, vol. 19, no. 15, p. 3420, 2019.
- [74] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

- [75] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: <http://arxiv.org/abs/1502.03167>
- [76] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in Statistics*. New York, NY, USA: Springer, 1992, pp. 196–202.
- [77] H. Hofmann, H. Wickham, and K. Kafadar, "Value plots: Boxplots for large data," *J. Comput. Graph. Statist.*, vol. 26, no. 3, pp. 469–477, 2017.
- [78] H. Kwon, G. D. Abowd, and T. Plötz, "Handling annotation uncertainty in human activity recognition," in *Proc. 23rd Int. Symp. Wearable Comput. (ISWC)*, 2019, pp. 109–117.



MARTIN GJORESKI received the B.S. degree in computer science from the Faculty of Computer Science and Engineering, Skopje, Macedonia, in 2014, and the M.S. degree in computer science from the Jožef Stefan Postgraduate School, Ljubljana, Slovenia, in 2016, where he is currently pursuing the Ph.D. degree in computer science. He has been a Research Assistant with the Department of Intelligent Systems, Jožef Stefan Institute, Ljubljana, since 2014. His research interest includes development of machine-learning methods for monitoring human states using wearable sensors. He developed parts of the machine-learning algorithms that received the Sussex-Huawei Locomotion Challenge, in 2018 and 2019, respectively.



MATJAŽ GAMS (Member, IEEE) received the Ph.D. degree. He is currently the Head of the Department of Intelligent Systems, Jožef Stefan Institute, Ljubljana, and a Professor of computer science with the University of Ljubljana and the Jožef Stefan Postgraduate School. His research interests include intelligent systems, artificial intelligence, cognitive science, intelligent agents, electronic and mobile health, business intelligence, and information society. He is a member of several international program committees of scientific meetings, national, the European Strategic Boards and Institutions, and editorial boards of 11 journals. He is also a member of the National Council of Slovenia, representing the field of science, for a period of 2017 to 2022. His team received four activity recognition competitions and placed in the finals of the XPrize Tricorder Competition. He is the Managing Director of journal *Informatica*.



MITJA LUŠTREK (Member, IEEE) received the Ph.D. degree in computer and information science from the University of Ljubljana. He is currently the Head of the Ambient Intelligence Group, Department of Intelligent Systems, Jožef Stefan Institute, Slovenia. His research interests include applying machine learning to sensor and other data in the domains of health, wellbeing, and ambient assisted living. His team received four activity recognition competitions and placed in the finals of the XPrize Tricorder Competition.



PELIN GENC received the B.Sc. and M.Sc. degrees in mechanical engineering from Middle East Technical University, Ankara, Turkey. She is currently pursuing the master's degree in informatics with Friedrich-Alexander University Erlangen-Nuremberg, Erlangen. From 2017 to 2019, she was a Research Assistant with Fraunhofer IIS. Her research interests include multimodal stress recognition, with a special focus on facial expressions, physiological signals, and psychophysiology of emotions and stress.



JENS-U. GARBAS received the Diploma and Ph.D. degrees in electrical engineering from Friedrich-Alexander-University Erlangen-Nuremberg, in 2004 and 2010, respectively.

In 2010 he joined Fraunhofer IIS. He was the Head of the Intelligent Systems Group, in 2011, and the Deputy Head of the Electronic Imaging Department, in 2012. Since 2020, he has been the Head of the Department Digital Sensory Perception. He is responsible for industrial and public research projects and software licensing in the areas of real-time computer vision, affective computing, and facial analysis.



TEENA HASSAN received the Bachelor of Technology degree in computer science and engineering from the Cochin University of Science and Technology, Kerala, India, and the Master of Science degree in autonomous systems from the Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany, in 2014. She is currently pursuing the external Ph.D. degree with the University of Bamberg. She continued her research in the field of automatic facial expression analysis with Fraunhofer IIS, Erlangen, Germany. She is currently a Research Associate with Bielefeld University and working in interaction architectures for social robots. Her research interests include modeling of facial muscle movements, fusing multiple sources of facial expression information, and modeling uncertainty in observations. Her contributions in this article are part of her research with Fraunhofer IIS.

...

5.3 Locomotion Recognition

Information Fusion 62 (2020) 47–62



Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus



Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors[☆]



Martin Gjoreski^{a,b,*}, Vito Janko^{a,b}, Gašper Slapničar^{a,b}, Miha Mlakar^a, Nina Reščič^{a,b},
Jani Bizjak^{a,b}, Vid Drobnič^a, Matej Marinko^a, Nejc Mlakar^a, Mitja Luštrek^{a,b}, Matjaž Gams^{a,b}

^a Jožef Stefan Institute, Department of Intelligent Systems, Ljubljana, Slovenia

^b Jožef Stefan Postgraduate School, Ljubljana, Slovenia

ARTICLE INFO

Keywords:

Activity recognition
Machine learning
Deep learning
Ensembles
Hidden markov models
Competition

ABSTRACT

The Sussex-Huawei Locomotion-Transportation Recognition Challenge presented a unique opportunity to the activity-recognition community to test their approaches on a large, real-life benchmark dataset with activities different from those typically recognized. The goal of the challenge was to recognize, as accurately as possible, eight locomotion activities (*Still, Walk, Run, Bike, Car, Bus, Train, Subway*) using smartphone sensor data. This paper describes the method we developed to win this challenge, and provides an analysis of the effectiveness of its components. We used complex feature extraction and selection methods to train classical machine learning models. In addition, we trained deep learning models using a novel end-to-end architecture for deep multimodal spectro-temporal fusion. All the models were fused into an ensemble with the final predictions smoothed by a hidden Markov model to account for temporal dependencies of the activities. The presented method achieved an F1 score of 94.9% on the challenge test data. We tested different sampling frequencies, window sizes, feature types, classification models and the importance of stand-alone sensors and their fusion for the task. Finally, we present an energy-efficient smartphone implementation of the method.

1. Introduction

The grand vision of wearable computing is that our devices will know as much as possible about our context in order to provide the best possible service. Which activity is being performed at any given moment is certainly an important part of our context, which is why Activity Recognition (AR) is intensely researched. Most research in AR is focused on the human body, dealing with activities such as walking, sitting and lying. However, transportation studies show that the average commute time is up to 80 minutes a day [1], and thus recognition of transportation modes can be as important as recognition of body-related activities.

The Sussex-Huawei Locomotion-Transportation (SHL) dataset [2] addresses this problem by providing a mixture of both activity types – containing eight different activities: *Still, Walk, Run, Bike, Car, Bus, Train, Subway*. In addition, the SHL dataset provides a unique opportunity for researchers to test their AR approaches against a common, real-life, large-scale benchmark dataset collected over a period of four months, which allows for cross-study comparison and systematic advancement of the research field. To promote the dataset, a competition called the “SHL challenge” was organized. The activities were to be recognized using seven smartphone inertial sensors: accelerometer,

gyroscope, magnetometer, linear accelerometer, gravity, orientation and barometer.

This paper offers a detailed description of our approach, which we used to win the SHL challenge by most accurately predicting the activities on an unlabelled dataset. In Section 2 we present the related work and in Section 3 we describe the SHL dataset, including data ordering and splitting. The next four sections present our method, which is schematically shown in Fig. 1. Section 4 starts with how the data was down-sampled and cut into windows, and how we derived additional data streams from the existing ones. Then it shows how we extracted a large body of features from every data stream (altogether calculating 1696 features), and it ends up with the three-step feature selection process that we used to select the best-performing feature subset. Section 5 presents both classical and deep learning (DL) algorithms that we trained for the AR. The classical models used the selected features, while the DL model was trained on the raw data. For the DL we used a novel architecture: an end-to-end multimodal spectro-temporal ResNet (Multi-ResNet). All trained models were combined into an ensemble that we used to generate predictions.

Section 6 describes how we used a hidden Markov model (HMM) to account for the temporal dependencies of sequential activities. Section 7 discusses an energy-efficient smartphone implementation

[☆] The first two authors should be regarded as joint first authors.

* Corresponding author.

E-mail address: martin.gjoreski@ijs.si (M. Gjoreski).

<https://doi.org/10.1016/j.inffus.2020.04.004>

Received 23 April 2019; Received in revised form 15 January 2020; Accepted 12 April 2020
Available online 21 April 2020

1566-2535/© 2020 Elsevier B.V. All rights reserved.

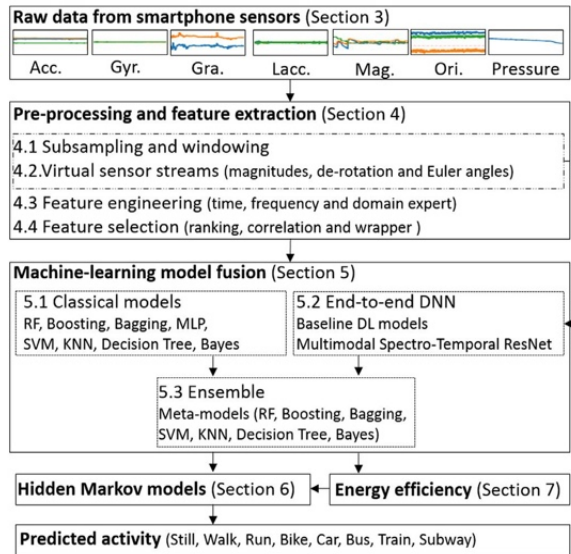


Fig. 1. Scheme of our method.

of our method. It proposes three different ways to increase energy-efficiency while retaining classification accuracy. Section 8 presents the experimental results including an extensive analysis on how different hyper-parameters of our method affect its accuracy. These include selecting which sensors to use, data sampling frequency (e.g., 100 Hz, 50 Hz or 10 Hz), window sizes (e.g., 60 seconds, 20 seconds or 10 seconds), feature types (e.g., time-domain or frequency-domain), feature selection methods (e.g., wrapper or ranking methods), machine learning algorithms (e.g., deep or classical, end-to-end or feature-based models), etc. Finally, in Section 9 we summarize our findings and discuss the limitations of the method and our future work.

The main contributions of the paper are:

- A comprehensive method for AR, which was validated by winning the SHL challenge.
- A novel end-to-end DNN architecture (Multi-ResNet) proven suitable for AR, a domain where DNNs are not yet established.
- An extensive analysis of the method's hyper-parameters, which can facilitate adaptation to other problems.
- Optimization of the method's energy-efficiency for a smartphone application.

In summary, we believe the proposed method is a solid baseline for other authors who attempt to work on the SHL dataset and similar AR problems.

2. Related work

2.1. Classical AR methods

The AR domain has been thoroughly explored in the past using body-worn sensors, ambient sensors and combinations of both. Here, we focus only on the body-worn sensors commonly found in smartphones, since those are most suitable for AR today due to the omnipresence of smartphones. The most frequent AR task is classifying activities in relation to movement, e.g., walking, running, standing still and cycling [3,4]. Most of the multimodal (multi-sensor) solutions use classical ML algorithms (e.g., Random Forest - RF, Support-vector machines - SVM, and k-nearest neighbours - KNN) to build models from features that are extracted from each modality independently. These models can be built using fusion in the feature space, i.e., all features are treated uniformly

[5,6]. There also exists fusion in the decision space, where models are built for each feature type with respect to the modality, and the final decision is output by a meta-model [7,8].

2.2. Deep AR methods

In the recent years, many attempts at AR were also made with DL end-to-end architectures [9,10]. The focus was on convolutional networks (CNN), which can automatically capture hierarchical feature representations of the data due to their stacked filtering layers [11]. CNNs, in combination with subsampling layers (e.g., pooling layers) and fully connected layers, are a very powerful end-to-end learning architectures [12–14]. In 2016, Ordonez et al. introduced the DeepConvLSTM [15], an architecture that stacks CNN layers over a multimodal long short-term memory (LSTM) recurrent neural network (RNN) for AR [16]. The LSTM layers allow for the model to learn the temporal dynamics in the input data by utilizing specific gates. Unlike HMMs, which are modelled on the Markovian assumption and have a finite number of hidden states, LSTMs have the advantage of a continuous space memory, which theoretically allows them to base their predictions on arbitrarily long past observations [17]. Most recently, Murahari et al. [18] experimented with attention mechanisms in the DeepConvLSTM, which improved its performance in the AR domain by a few percentage points.

2.3. Transportation recognition methods

The approaches that recognize transportation modes utilize very similar techniques to the standard AR approaches. For example, Martin et al. [19] developed a method for real-time prediction of the transportation mode using smartphone GPS and accelerometer data. They combined dimensionality reduction methods (PCA) and ML algorithms (KNN and RF) to accurately classify five modes of transportation (i.e., walking, biking, car, bus and rail). Fang et al. [20] developed a method for the recognition of transportation and vehicular modes using smartphone accelerometer, magnetometer and gyroscope. Their solution is based on decision trees, KNN and SVM. Reddy et al. [21] utilized a combination of a decision tree followed by a first-order discrete HMM to account for the temporal dependence between the labels. Similarly, Hemminki et al. [22] developed an accelerometer-based transportation recognition method for smartphones capable of recognizing six different modes (i.e., stationary, walking, bus, train, metro and tram). Their solution is based on a combination of AdaBoosting and a discrete HMM.

2.4. SHL Challenge methods

A comprehensive overview of all AR methods competing at the SHL challenge is presented in the summary paper by Wang et al. [23]. Overall, there were nineteen submissions, of which eleven used only classical AR methods, and eight either used only DL or included DL methods. From the top five methods, the teams “S304” and “Confusion Matrix” achieved a similar F1 score of 87.5%. “Confusion Matrix” used an RF model and then smoothed the estimation with majority voting [24]. S304 used a multi-layer perceptron neural network and then smoothed the estimation with an HMM [25]. The third placed team “Tesaguri” achieved an F1 score of 88.8% by applying CNNs to the spectrogram of the sensor data [24]. Our two teams “JSI-Deep” [26] and “JSI-Classic” [27] were the only two teams that achieved an F1 score over 90%. More specifically, JSI-Classic achieved an F1 score of 92.4%, by combining tree-based XGBoost with advanced feature extraction and feature selection techniques. “JSI-Deep” achieved the highest F1 score of 93.9% by using a meta method that utilizes both classical and spectrogram-based end-to-end deep learning methods merged in an ensemble whose final predictions are smoothed by a discrete HMM. This study presents a detailed analysis of the “JSI-Deep” and “JSI-Classic” methods with a few additional novelties including a novel spectro-temporal end-to-end DL

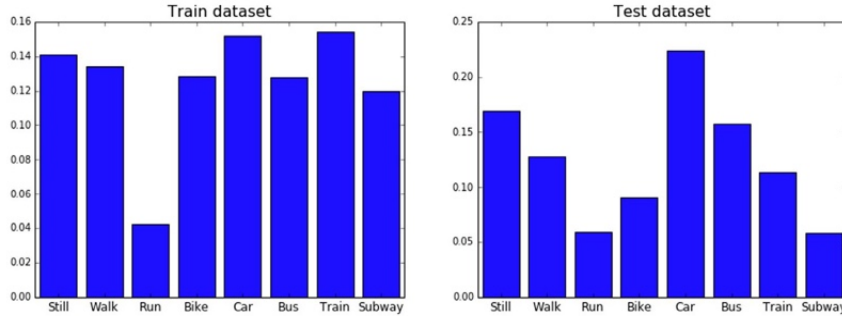


Fig. 2. Activity distribution of the *train* and *test* datasets.

architecture, and a methodology for making our solution more energy-efficient at the cost of a small decrease in accuracy. The “updated JSI-Deep” method presented in this study, achieved an F1 score of 95.2% on the challenge test data.

3. SHL challenge dataset

3.1. General description

The SHL Dataset is one of the largest AR datasets, suitable for a wide range of studies in fields such as transportation recognition, AR, mobility pattern mining, localization, tracking and sensor fusion [2]. It was recorded over a period of 7 months by 3 participants engaging in 8 different activities in a real-life setting in the United Kingdom. A subset of this dataset was used for the SHL Challenge and was subsequently used throughout this work. The dataset for the SHL challenge is available on the challenge website [28] and is described in detail in the baseline paper [29].

All of the data was provided by a single smartphone, worn by the same user in his trouser pocket. Data came in five different modalities: acceleration, magnetic field, angular velocity (measured by gyroscope), air pressure and the phone’s current orientation. The acceleration was broken down into two additional data streams: the acceleration as the result of gravity, and the acceleration without the gravity component (linear acceleration). Notably, sensors that could identify the user’s location – GPS, Wi-Fi, cell network – were not included.

The competitors were provided with a *train* and a test (*SHLtest*) set. The former came with labelled activities, while the labelling of the latter was the competition’s goal. However, by the time of writing of this paper, the *SHLtest* set labels were publicly released, allowing us to present the results on both data sets. All together, there were 82 recorded days (62 for training, 20 for testing). This provided 16,310 minutes, or roughly 272 hours, of data.

There were 8 activities of interest: *Still*, *Walk*, *Run*, *Bike*, *Car*, *Bus*, *Train*, *Subway*. The activity distribution in the *train* set (Fig. 2) was mostly uniform, with the exception of the *Run* activity, which was (understandably) under-represented. The *SHLtest* set, on the other hand, was skewed towards the *Car*, *Bus* and *Still* activities. To simulate the competition conditions, we treated this difference in the distributions as an unknown and made no steps to make them more similar.

Example samples of the acceleration and magnetic field data are given in Fig. 3. Both exhibit a periodic and distinct pattern with different magnitudes and frequencies when the user is either walking, running or cycling. Acceleration data also captures the noise present when driving in vehicles. In addition, the magnetic field sometimes wildly oscillates while the user is in the train or subway, which can be potentially exploited for recognizing them.

3.2. Data ordering and data splits

The recording scenario of the SHL dataset was relatively natural, with the test subject moving around the city for several days (as op-

Table 1
Average duration of each activity.

Activity	Still	Walk	Run	Bike	Car	Bus	Train	Subway
Length [mins]	14	12	12	20	60	34	64	34

posed to the scenarios typically performed and recorded in a lab). This resulted in activities that on average lasted a long time (Table 1), with sensible transitions from one to another (Fig. 4). Both properties were exploited for the HMM smoothing (Section 6) and for energy-efficiency optimization (Section 7).

The data was split into one-minute segments, which were shuffled by the dataset’s authors. The order of the one-minute segments was provided for the *train* set, but not for the *SHLtest* set. We assume this was done in order to enforce the use of a window length of less than or equal to one minute when classifying the data.

Our first step was to order the *train* set using the provided ordering. Then we split it into an internal validation set (*ivalid*), internal test set (*itest*) and internal train set (*itrain*) – using the first 25%, second 25% and last 50% of the *train* set, respectively. The *itrain* set was used to train our models. The *ivalid* set was used to select the appropriate features, sampling frequency and window size, and to train a meta-model. The *itest* set was used to estimate the performance of our model. All three sets were checked to ensure they roughly retained the same activity distribution as the original *train* set. We used the competition’s *SHLtest* set for the final evaluation of our method. Note that the ordering of the data before making the split for internal evaluation was a key step, as otherwise two subsequent data samples (which are usually very similar) would frequently be one in the *itrain* set and the other in the *itest* set – leading to model overfitting.

The ordering of the *SHLtest* set was not required for our base model; however, at least locally ordered data was needed for the additional HMM smoothing step in an attempt to further increase the classification performance. Therefore, we had to create a method that could reassemble the one-minute segments into the original order.

Intuitively, data at the end of one segment should closely match the data at the beginning of the next one. A simple procedure can therefore be applied:

- step 1: Select a random one-minute segment x .
- step 2: Find the one-minute segment y that has the minimum distance from its first sensor reading to the last sensor reading of x . The distance used is the Euclidean distance with different sensors weighted using empirically determined weights.
- step 3: If the minimal distance is lower than a predetermined threshold, assume that y follows x and repeat step 2 with y as the new x . If not, assume that the recording was interrupted and start assembling a new sequence of data repeating step 1.

On the *SHLtest* set, we produced 42 sequences of data that turned out (when the ordering was revealed after the competition) to be 99.8% correctly sorted, with roughly 10 instances out of place. The order of

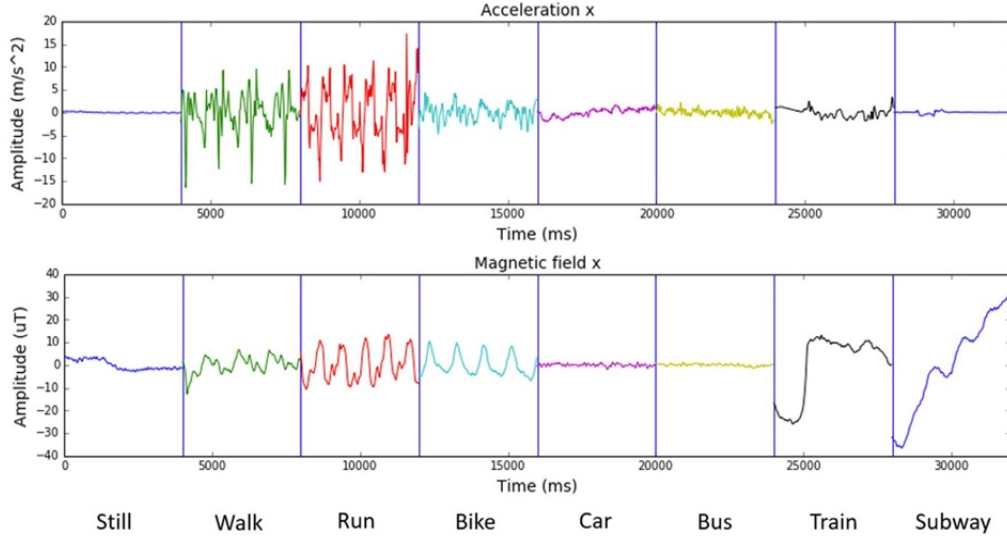


Fig. 3. Acceleration and magnetic field data samples (both taken from the x-axis). A four-second sample is shown for each activity.

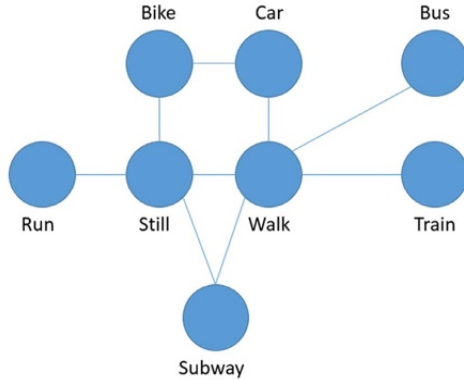


Fig. 4. All activities, connected if the probability of a transition from one to another is more than 30%.

the sequences themselves is largely irrelevant, however, as each had the HMM smoothing applied independently.

4. Pre-processing and feature extraction

This section describes the pre-processing step – windowing, down-sampling and deriving the virtual sensor streams – as well as the extraction of features used for training the classical ML models and their selection process. The DL methods use the raw data instead of features here described.

4.1. Data frequency and window size

The data was originally sampled using a 100 Hz sampling rate for all sensors. All sensor readings and activity labels coincided at the same timestamps. This frequency was previously empirically determined as more than adequate for most AR tasks [30,31]. However, the high sampling frequency presents two practical problems: 1) the further steps (notably the feature extraction) are slow due to the amount of data that needs to be processed; and 2) if this method would later be implemented

on a smartphone, the required sampling frequency and the constant processing would shorten the battery life of the device.

To mitigate these issues, we first down-sampled the data. To determine a sensible frequency, we tested different frequencies using two different algorithms: a simple RF using 100 trees and frequency domain features (which were fastest to calculate), and the Multi-ResNet DL model presented in Section 5.2. We chose the RF as the baseline method because it is robust to noise, relatively fast to train and frequently used in the AR domain.

The next decision was to select the appropriate window size, which is used to split the sensor data into windows which are either fed directly to the Multi-ResNet or used for the feature extraction for the classical ML algorithms. Longer windows naturally contain more data and are expected to enable greater classification accuracy. Shorter windows, on the other hand, can detect an activity change faster. We once again used the same algorithms (RF and Multi-ResNet) to test the different window sizes, and repeated these tests with different frequencies. The results of these experiments are shown in Section 8.1.

The activity label was calculated as the majority of the per-sample labels (with the frequency of 100 Hz) in each window. The same procedure was used for testing, calculating the *majority-label accuracy*. This method was used throughout this work, as it is convenient to calculate and is used in most other works in the AR domain. Alternatively, one can compare the model's predictions with the per-sample labels, calculating the *per-sample accuracy* – which was used by the SHL challenge organizers. The majority-label accuracy is in general slightly higher, but the differences are nearly negligible (Section 8.1).

4.2. Virtual sensor streams

The SHL dataset provides 20 different sensor streams, if we are individually counting each axis of the 7 provided sensors. From these original sensor streams it is possible to derive additional sensor streams that are useful for the AR. The subsequent steps treat these derived sensor streams like any of the original ones.

The first derived sensor stream is the magnitude Eq. (1) of the data. It was calculated for all the data coming from tree-axis sensors (acceleration, linear acceleration, gravity, magnetic field and angular velocity).

$$m = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

Second, the orientation data originally presented in the quaternions format $[q_w, q_x, q_y, q_z]$ was converted into the Euler angles – roll, pitch, yaw. While the quaternions have some desirable mathematical properties, the Euler angles are more intuitive and can be individually interpreted (one value for rotation around each axis). To make this transformation, we used the standard formulas for the task:

$$\text{pitch} = \arctan\left(\frac{2(q_w q_x + q_y q_z)}{1 - 2(q_x q_x + q_y q_y)}\right) \quad (2)$$

$$\text{roll} = \arcsin(2(q_w q_y - q_z q_x)) \quad (3)$$

$$\text{yaw} = \arctan\left(\frac{2(q_w q_z + q_x q_y)}{1 - 2(q_y q_y + q_z q_z)}\right) \quad (4)$$

Third, additional sensor streams were created by rotating the accelerometer and magnetometer data from the phone's coordinate system to the “world” (North-East-Down) coordinate system. This could be useful for determining, for example, if the magnetic field is coming from above or below, as the same axis is always pointed upwards. The transformation was done by multiplying the current values Eq. (6) with the coordinate system change matrix Eq. (5), using quaternions to determine the current orientation [32].

$$R_{NB} = \begin{bmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_x q_y + q_w q_z) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z - q_w q_x) \\ 2(q_x q_z - q_w q_y) & 2(q_y q_z + q_w q_x) & 1 - 2(q_x^2 + q_y^2) \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{world}} = R_{NB} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{sensor}} \quad (6)$$

In total we added fourteen derived data streams – five magnitudes, three Euler angles and six streams for rotated acceleration and magnetic field data (three axes each).

4.3. Feature engineering

Features were individually calculated for each data stream, with some features using data from all three axes of the same sensor at the same time.

Features can be roughly split into two categories: frequency-domain and time-domain. In aggregate, a total of 1696 features were computed and used in the subsequent steps.

4.3.1. Frequency-domain features

These features were calculated using the power spectral density (PSD) of the signal, which is based on the fast Fourier transform (FFT). PSD characterizes the frequency content of a given signal and can be estimated using several techniques. The simplest one is to use a periodogram, which is obtained by taking the squared-magnitude of the FFT components. An alternative to a simple periodogram is Welch's method, which is also widely used and commonly considered superior to the periodogram. It computes the average of the periodograms of multiple overlapping segments of the signal to reduce the variance of the PSD. In our work, we opted to use Welch's method to obtain the PSD.

Using PSD is only suitable when the signal is clearly periodic. We chose to test windows of length ranging from 5 to 60 seconds. These were long enough to contain several periods of human motion as well as vehicle vibration. Sample periodical patterns can be clearly seen in Fig. 3 for the activities: Walk, Run, Bike, Car and Bus.

We implemented frequency-domain features as given in related work [33]. The following features were computed.

- **Three largest magnitudes.** Three peaks with the largest magnitude from the PSD were considered. These tell us the dominant frequencies in the signal. Both the magnitude values and the frequencies (in Hz) were taken as features.

- **Energy.** Calculated as the sum of the squared FFT component magnitudes. The energy was then normalized by dividing it with the window length.

$$\text{energy} = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2, \quad (7)$$

where $x(n)$ is the n -th FFT component and N is the parameter specifying the number of FFT components to compute.

- **Entropy.** Calculated as the information entropy of the normalized FFT component magnitudes. It helps in discriminating between activities with a similar energy feature.

$$\text{entropy} = - \sum_{n=0}^{N-1} x(n) \log(x(n)) \quad (8)$$

- **Binned distribution.** A normalized histogram, which is essentially the distribution of the FFT magnitudes into 10 equal sized bins ranging from 0 Hz to 25 Hz.

- **Skewness and kurtosis.** Calculated on the PSD. Skewness and kurtosis describe the shape of the distribution of the PSD. More precisely, skewness tells us about the symmetry of the distribution while kurtosis tells us about its flatness.

4.3.2. Time-domain features

As the time-domain features, we used the expert features previously used in our other works in similar domains [6,34], including one previously-won competition [3].

A description and analysis of the expert features can be found in our previous paper [6]. In summary, the magnitude data stream provided the information on the intensity of the activity, while the individual axes provided the information on the orientation of the device and subsequently on the position of the user.

Some features come from statistics and describe the intensity and shape of the signal: the mean, variance, Pearson's correlation between axes, their covariance, skewness, kurtosis, quartile values and range between them. Others have a more physics-based interpretation, such as velocity and kinetic energy. The rest came from expert knowledge of the domain: the number and height of peaks in the window, the signal's mean, its sum and squared sum, and the number of times the signal crosses its mean value.

In addition, we used a subset of features from the *tsfresh* library [35] that seemed interesting, but were not included in our set of expert features. These features were: the signal minimum, maximum, the number of times the signal is above/below its mean, the signal's mean change and its different autocorrelations (correlations of the signal with a delayed version of itself, for different delays).

Some of the features were calculated on the unfiltered data streams, while some were calculated on data filtered with either a simple low-pass or band-pass filter. The filters were calculated using Eqs. (9) and (10), where x represents the raw data and y represents the filtered data.

- **Low-pass filter.** Removes the low frequencies from the data. Useful for filtering out, for example, the gravity component of the acceleration.

$$y_i = \alpha x_i + (1 - \alpha) y_{i-1} \quad (9)$$

- **High-pass filter.** Removes the high frequencies from the data. Useful for filtering out noise.

$$y_i = \alpha x_i + \alpha(x_i - x_{i-1}) \quad (10)$$

- **Band-pass filter.** This filter simply applies both low and high pass filters one after another.

4.4. Feature selection

Given the relatively high number of features (1,696), we used a feature selection procedure to remove the features that do not contribute

to the accuracy of the model and only increase the odds for overfitting. Calculation of fewer features also contributes to the computational efficiency of our solution and simplifies its implementation. Our feature selection consisted of three steps.

In the first step, the mutual information [36] between each feature and the label was estimated using the *invalid* set, where larger mutual information means higher dependency between the feature and the label.

After the features were sorted according to this value, correlated features were removed based on the Pearson correlation coefficient [37]. This showed that roughly half of the features were redundant, which was expected due to the number of features and the similarity of the data streams. To make the process computationally feasible, only 100 features were taken at a time, starting with those with the highest mutual information with the label. Correlation was then calculated for each pair. If the correlation was higher than a certain threshold (experimentally determined as 0.8), the feature with lower mutual information was discarded. After that, the next 100 features were added to the remaining set and the correlation between each pair was calculated again.

In the final step, features were selected using a greedy “wrapper” algorithm. An RF model was first trained using only the feature with the highest mutual information. The trained model was used to predict labels for the *invalid* set, and the prediction accuracy was calculated. Then the second-best feature was added and the model was trained again. If the accuracy on the *invalid* set was higher than without using this feature, the feature was kept. This procedure was repeated for all remaining features. This strict selection initially led to overfitting to the *invalid* set (accuracy was much higher compared to the *itest* set), so the condition for keeping a feature was made less strict: the feature was kept if the accuracy did not decrease by more than an experimentally set improvement threshold. Using this rule, overfitting to the *invalid* set was reduced. The results after each feature selection step are shown in Section 8.2.

5. Classification models

Different classical ML algorithms and an end-to-end DL architecture were trained and tested. After training the individual models, we combined them in an ensemble which further increased the classification performance. In this section we describe the models, and in Section 8.3 and Section 8.4 show the results.

5.1. Classical models

For the classical models, we used the ML algorithms as implemented in the scikit-learn ML toolkit [38]. Each of these algorithms used features as selected in Section 4. The models were tested both with the default hyper-parameter values and with those found using a 2-fold randomized parameter search. The randomized parameter search was performed using the following procedure: for each algorithm, parameter settings were randomly sampled from distributions predefined by an expert. Next, models were built with the specific parameters and then evaluated using a 2-fold validation procedure on the train data. This search was repeated 10 times using different folds. In Section 8.3 we report the averaged results, but we kept the best-performing parameters for the final models. We experimented with the following ML algorithms: Decision Tree [39], RF [40], Naïve Bayes [41], KNN [42], SVM [43], Bagging - using Decision Trees [44], Adaptive Boosting (AdaBoost), Extreme Gradient Boosting (XGB) and Multilayer Perceptron (MLP) [45], label switching both with MLPs [46] and with DTs [47].

5.2. Deep learning models

We propose a deep multimodal spectro-temporal ResNet (Multi-ResNet) and describe it in Section 5.2.2. To compare its performance to simpler DL architectures, we describe and test them as well.

5.2.1. Baseline deep learning models

To provide an end-to-end baseline comparison for our Multi-ResNet, we implemented five DL networks: a network with four CNN layers followed by two fully connected layers to output the final activity prediction (CNN-4); two similar networks with eight and sixteen CNN layers (CNN-8 and CNN-16); one network with two LSTM layers followed by two fully connected layers (LSTM-2); and one ConvLSTM mimicking the architecture proposed by Ordonez et al. [15], CNN-4-LSTM-2. Each CNN layer is followed by a batch normalization layer for reducing internal covariate shift [48], and a ReLU activation layer [49], which speeds up the training process compared to other activation layers (e.g., tanh). To avoid overfitting, L2 regularization and dropout was used for the dense layers. The training of the networks was fully supervised, by back propagating the gradients through all the layers. The parameters were optimized by minimizing the cross-entropy loss function using the Adam optimizer. The models were trained with a learning rate of 10^{-4} and a decay of 10^{-4} . The batch size was set to 1024 and the maximum number of training epochs was set to 75. The models were trained with early stopping on the *invalid* data.

5.2.2. Multimodal spectro-temporal resnet

Our deep multimodal spectro-temporal ResNet (Multi-ResNet) is depicted in Fig. 5. The structure is based on the idea by He et al. [50] for training very deep end-to-end networks for image recognition by using shortcut (residual) connections. Using the residual networks, Wang et al. [51] proposed an end-to-end unimodal time-series classification network. Our network builds upon Wang’s work with two additional novelties, which are key factors for a successful AR system, i.e., multimodal and spectro-temporal information fusion.

For each sensor channel, the network extracts channel-specific spectro-temporal information: the spectral information is extracted by calculating the spectrogram in decibels, i.e., $\log_{10}$ of the amplitude spectrogram, for each input window; the temporal representation is extracted by the residual blocks that contain CNN layers with 1-dimensional filters. The shortcut connections in the residual blocks combat the gradient vanishing problem, i.e., the more layers there are, the smaller learning update each layer receives, thus the harder the training is [30]. For example, for a 20 channel input (3 accelerometer, 3 gyroscope, 3 gravity, 3 linear acceleration, 3 magnetometer, 4 orientation and 1 pressure channel), 4 residual blocks per channel and 3 CNN layers per residual block, the network will have 240 CNN layers ($20 \times 4 \times 3$) through which the gradient needs to be propagated, thus we need a mechanism to avoid the gradient-vanishing problem [52]. Each CNN layer is followed by a batch normalization layer, and a ReLU activation layer. Each residual block ends up with an average pooling layer which is used for dimensionality reduction. The output of the channel-specific layers, i.e., channel-specific spectro-temporal information, is then fused by two dense (fully connected) layers. L2 regularization and dropout was used for the dense layers. The final output of the Multi-ResNet is provided by a softmax layer, which represents a class probability for each of the eight different classes. The network was trained using the same learning procedure (loss function, optimizer, learning rate, decay, batch size and early stopping) as the baseline DL models.

The original “JSI-Deep” method that won the SHL challenge used only the spectrogram part of the Multi-ResNet, i.e., the spectrograms were calculated for each sensor channel and were fed into fully connected layers, which provided the final predictions [26]. In this paper, that architecture is referred to as DNN-Spectrogram, in order to provide comparison between the two DNN architectures (Multi-ResNet vs. DNN-Spectrogram).

5.3. Ensemble

Finally, we experimented with an ensemble of models built using a stacking approach. The base models in the stacking ensemble are the classical models and the Multi-ResNet, described in the previous two

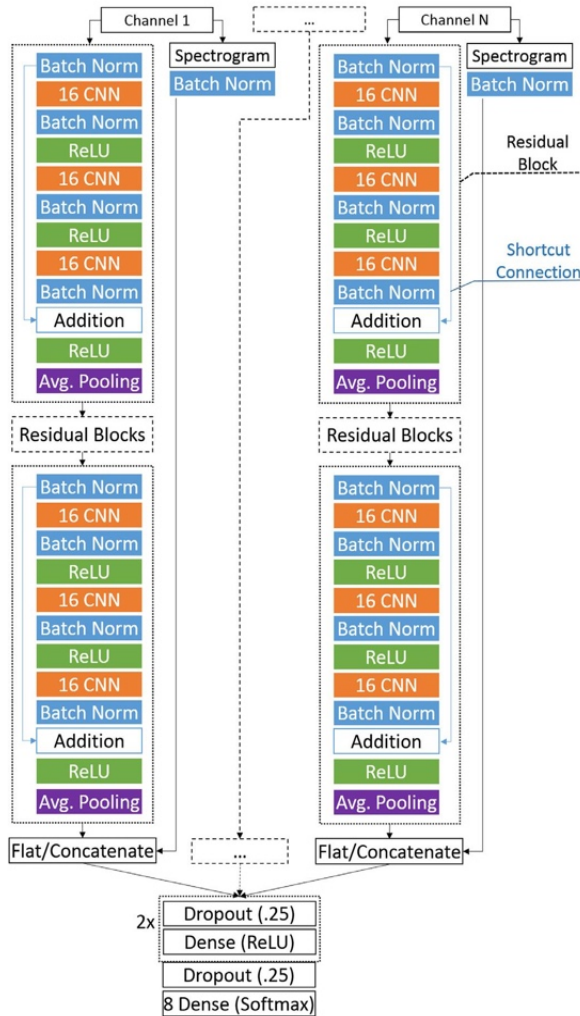


Fig. 5. Proposed Deep Multimodal Spectro-Temporal ResNet (Multi-ResNet).

subsections. The meta learner is a model that takes as input the outputs of base models. We evaluated meta-models built with the different ML algorithms: RF, Gradient Boosting, XGB, AdaBoosting, SVM, KNN, Gaussian Naïve Bayes and Decision Tree, and tuned the hyper-parameters. Each base model was trained on the *itrain* set and each meta-model was trained on the base models' output for the *ivalid* set. The hyper-parameter tuning was again performed using a 2-fold randomized parameter search on the train data.

6. Hidden markov model smoothing

In all experiments so far, all the windows were classified independently from one another. This approach discards all the information on temporal dependencies between them. If a user is currently on a train, for example, but the next window is classified as *Bus*, followed by another *Train* classification, it is far more likely for *Bus* to be a misclassification than a vehicle switch (Fig. 6).

This motivated us to use an extra step after classification, where the temporal information was taken into account. This was done using an HMM model. In this model (a small part of it shown in Fig. 7) the hidden

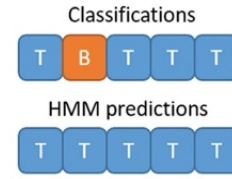
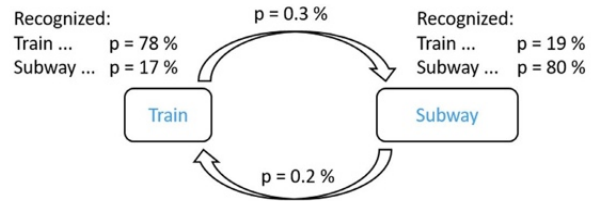
Fig. 6. Top row shows a sequence of *Train* and *Bus* classifications. They are corrected using HMM smoothing into a sequence of only *Train* activities shown below.

Fig. 7. A small part of the HMM model, where the hidden states represent true activities and the visible states are the recognized activities.

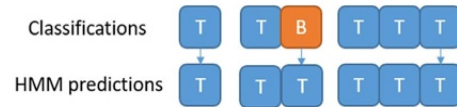


Fig. 8. Iterative HMM predictions. The top row represents the classified activities, and the bottom one represents the activities predicted by the HMM smoothing. Each group represents one step. In each step, the sequence is iteratively lengthened and only the last instance is corrected.

states represent the actual activity, while the visible output represents the classified activities.

The parameters of this model are the transition probabilities between the states and the probabilities of observed emissions in each state. The former can be estimated from the transition matrix of the *itrain* set (matrix of probabilities that one activity is followed by another), while the latter from the confusion matrix of the *ivalid* set.

The HMM smoothing was performed using the Viterbi algorithm in the *hmmlearn* library [53]. This algorithm uses all the available data for “correcting” each instance. From the viewpoint of a single instance, it means that all of its predecessors and successors must be known in order to “correct” it. This is, however, impractical for a real-time monitoring application implemented on a smartphone.

Having the past predictions corresponds to saving predictions as they are made, while having the future predictions means that the HMM smoothing must happen with a delay. To test the usefulness of the HMM smoothing in a practical setting, we tested how different delays (and having no delay – Fig. 8) affect the accuracy of the HMM predictions. To “correct” an instance before these tests, the algorithm was run only on the predictions that happened before the instance and on a number of instances (equal to the delay) that happened after it. The results are shown in Section 8.5.

7. Energy-efficient solutions

The previous sections were focused on how to achieve as high an accuracy as possible using a meta-model in combination with the HMM smoothing. To do so, we used all the available data from every sensor. In practice, however, turning on all the sensors of a smartphone will quickly drain the battery of the device, making the application undesirable from the user's point of view. It would thus be beneficial to find

solutions that are almost as good in terms of accuracy, but are much more energy-efficient.

There are four most frequently used methods to reduce the energy consumption: using only a sensor subset, periodically turning the sensors on and off (*duty-cycling*), reducing the sampling frequency and reducing the processing power required for the classification. In this section we briefly describe – and apply on the SHL dataset – a general methodology that can quickly create energy-efficient solutions using the first two methods without any expert knowledge of the domain. This methodology is described in detail in our previous works [54,55]. The last two optimization modalities (sampling frequency and processing power required) are briefly discussed, using simple heuristics, at the end of the section.

Trying to estimate the energy consumption of a smartphone device can be surprisingly difficult, as: 1) it is heavily device-dependent; and 2) energy consumption of different components do not add up linearly. For example: accelerometer and gyroscope active together typically consume less energy than the sum of their individual consumptions. In order to avoid both complications, we simply measured what proportion of the original data we needed for a particular solution. This is a loose estimate of the energy efficiency, as the more data a system needs, the more energy it must consume to both collect and process it. This simplification is also appropriate because all sensors present in this dataset individually consume roughly the same amount of energy. Note, however, that the same methodology could be used with real energy estimates [56,57] if one would be interested in a particular device.

The HMM smoothing cannot be directly integrated into the methodology, so we omitted this post-processing step when searching for the solutions and then used it only on solutions found. Additionally, we used the ensemble without the DL model, as it can be impractical to use it on a smartphone. While recent research [58,59] show that DL models can be adapted to make them more resource efficient and thus more suitable for smartphones, we left such adaptations of our models to future work.

7.1. Choosing sensor subset

Instead of constantly using all five sensors, using only a subset of them would increase the energy-efficiency of the system. The individual performances of all sensors are listed in Table 6. Deriving a table of all possible sensor combinations ($2^5 = 32$) is likewise simple. Doing so reveals that individually the accelerometer is the sensor with by far the highest accuracy, and that the accelerometer with the magnetometer make the best pair. It shows that the contribution of other sensors rapidly falls afterwards, but that they are all required for achieving the highest possible accuracy.

Further improvements can be made by optimizing the sensors for each activity. For example, the magnetometer and barometer might only be useful in vehicles but not while walking or running. We can therefore create an assignment from each activity to a different sensor set, and whenever an activity is classified, the corresponding sensors are turned on. Since the activities last for minutes, the cost of switching a sensor on/off is minimal.

The problem of finding the ideal sensor-subset-to-activity assignment is not trivial. Not only are there $(2^5)^8 \approx 10^{12}$ different assignments, given 5 sensors and 8 activities, but it can be time consuming to test an individual one, as one has to go through the whole dataset, classifying each instance and switching sensor streams and models in the process. Note that trying to circumvent this by testing how a sensor subset works for recognizing an activity, and then aggregating this performance for every activity, makes a poor approximation. In practice, a misclassification can turn on sensors that are inappropriate for the current activity, leading to further errors.

To deal with both problems (the number of assignments and their evaluation) we used the method from our previous work [54]. In short, it uses the *steady-state* of a Markov-chain model to predict how would an assignment behave in terms of both energy and accuracy. To calcu-

late the parameters of the Markov-chain model, it uses the matrix of transition probabilities between the activities and the different models (that use different sensor subset) confusion matrices. It then uses a genetic multi-objective algorithm (NSGAA-II [60]) – the objectives being energy consumption and accuracy – evaluating every assignment with the Markov-chain model. This combination can quickly search through the problem space and efficiently find a set of non-dominated solutions.

Doing so generates an approximation for the Pareto front of solutions, each solution representing a different trade-off between accuracy and energy consumption. From this set of trade-offs, a system designer can pick one that is suitable for his application requirements. Section 8.7 presents the Pareto front for the SHL dataset and some sample solutions from it.

7.2. Duty-cycling

As shown in Table 1, most of the activities are relatively long-lasting. Therefore turning the sensors off when an activity is recognized and then turning them back on a few minutes later could be a way to further optimize the energy consumption of the system. The system simply assumes that the activity has not changed while the sensors were turned off. This process, called *duty-cycling*, can similarly to before, be enhanced by optimizing the cycle length for each activity individually. Intuitively, long lasting activities should have longer cycles, and vice versa. Additionally, if an activity is often followed by a short-lasting activity, the duty-cycle length should also be shorter, so as not to miss the following short activity entirely. Finally, one has to consider the effect of misclassifications, as misclassifying an activity might subsequently cause a wrong duty-cycle length.

This once again introduces two problems: 1) combinatorial complexity, as there are many possible duty-cycle-length-to-activity assignments – l^8 , if the maximum cycle length is l ; 2) modelling the performance of the system given a duty-cycle-length-to-activity assignment. We solve both problems by using our previous work [55], where we once again use Markov-chain calculus (although in an altogether different way) to determine the effect of an assignment on the system performance and then use the NSGAA-II algorithm to find a set of non-dominated solutions.

In Section 8.7, we show sample solutions where we used the same duty-cycle length for all activities, and we compare them to solutions where activity-specific duty-cycle lengths are used.

7.3. Combination

In our previous works, we used both described methods independently. In this work, we combine them, by first selecting which sensor subset to use for each activity and then running the duty-cycle optimization using this assignment. This requires two choices (one in each step) regarding which solution from the Pareto front to use, but otherwise seamlessly combines the two.

One can combine the selected solution with the optimization of the sampling frequency – which is the third, smaller contributor to the energy consumption of the device. Testing different frequencies (Section 8.1) revealed that using the frequency of 50 Hz loses basically no classification accuracy compared to the frequency of 100 Hz. We thus settled to using it, and made no additional effort at optimizing the frequency, but a reader can find more sophisticated methods in related work [30,61].

Finally, to reduce the power consumption required for classifying a single instance, we compared the processing requirements of different classifiers and their ensembles. Different trade-offs between the classification accuracy and processing power required are shown in Section 8.7. It should be noted, however, that since classifications happen only once each minute, the data processing is not the bottleneck when considering the energy consumption.

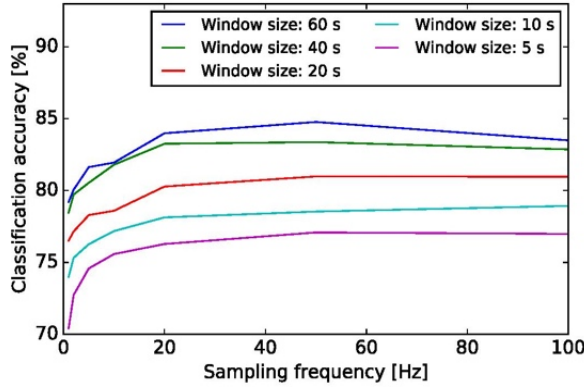


Fig. 9. Accuracy on the *itest* set for different frequencies and for different window sizes, using the RF model.

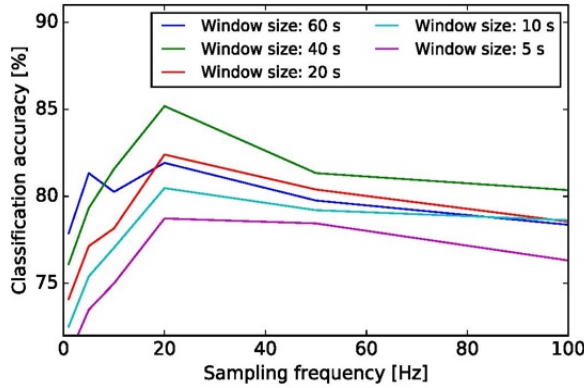


Fig. 10. Accuracy on the *itest* set for different frequencies and for different window sizes, using the Multi-ResNet.

8. Results

For the experimental evaluation, we mostly used classification accuracy as the measure of classification quality, but for the final results we also list the precision, recall, and F1 score:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{All Test Instances}} \quad (11)$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (13)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (14)$$

When calculating the F1 score for multiple activities, the F1 score is calculated for each activity separately and then averaged over all of them.

8.1. Frequency and window size

We started by testing different sampling frequencies and window sizes. Results using two different models (trained on the *itrain* set and tested on the *itest* set) are shown in Figs. 9 and 10.

The results for the RF show almost no difference in performance when dropping from a sampling frequency of 100 Hz to a sampling frequency of 20 Hz. At lower frequencies, the performance started to visibly decrease, but surprisingly, even at a sampling frequency of 1 Hz,

Table 2

Difference in accuracy on the *itest* data using the per-sample labels and the majority labels, using different frequencies and window sizes.

Window length [s]	60	40	20	10	5
Difference [%] at 100 Hz	0.57	0.14	0.04	0.01	0.02
Difference [%] at 1 Hz	0.55	0.17	0.04	0.01	0.01

Table 3

Accuracy [%] on the *itest* data achieved using data from only one sensor.

Sensor	Acc	Gyro	Pressure	Mag	Ori	All
RF	86.5	79.5	52.1	72.1	71.3	82.6
Multi-ResNet	81.4	67.2	49.0	67.7	65.9	85.2

the accuracy only fell by roughly 5 percentage points compared to the highest value. Given the results in similar domains, it is not surprising that 20 Hz is enough to recognize the human motion, however it is interesting to note that it also seems enough to capture the different vehicle vibrations. Additionally, we can see that larger window sizes increase the accuracy regardless of the frequency.

The results for the Multi-ResNet show that the accuracy reached its peak at a sampling rate of 20 Hz and decreased at lower rates, just as in the RF case. Somewhat curiously, the 40 second window was found the best performing by a significant margin, while it was a close second in the RF case.

For the classical ML models, we selected a sampling rate of 50 Hz and a window size of 60 seconds, to match our submission to the SHL competition. However, these tests show that any other frequency from 20 Hz to 100 Hz would be similarly suitable. For the Multi-ResNet, we continued with the best performing parameter combination, i.e., a sampling rate of 20 Hz and a window size of 40 seconds, since the accuracy significantly decreases for other combinations.

Next, we investigated the difference between having per-sample labels and majority labels. To do so, we calculated the accuracy for both cases for each window size. These tests were done using two frequencies at the opposite end of the spectrum. The results are presented in Table 2, and show that the difference is (sometimes significantly) less than 1%. Results were also unaffected by the selected frequency. Due to the small difference and the reasons explained in Section 4.1, the results reported from here on will continue using the majority labels.

Lastly, we tested the sensitivity to the choice of the *itrain*, *itest*, *invalid* set. To do so, we tested what happens if we switch them around, e.g., training on the *itest* and testing on the *itrain*. All the experiments from Fig. 9 were repeated for all six possible switches of these three sets. The standard deviation of the results was 1.6% – from which we can conclude that the dataset is quite robust to the choice of the internal train, test, and validation sets. This can probably be attributed to its relatively large size.

8.2. Feature importance and selection

The next area for attention is the importance of the features based on both the sensor stream they were derived from and their type (time-domain, frequency-domain, rotated to world's coordinate system etc.)

We start by training the models on data from only one sensor stream, using both the RF and the Multi-ResNet as before. Results (Table 3) show – somewhat unsurprisingly – that the accelerometer is the best suited for the task, with the gyroscope and magnetometer data in second and third places. For the Multi-ResNet, no sensor stream by itself outperforms their combination – showing the importance of the sensor fusion in this domain. In the RF case, the accelerometer seems to outperform the sensor union, but this ceases to be the case after the feature selection step.

Table 4

Accuracy [%] on the *itest* data for different feature subsets using RF.

Features	Mag	Acc
Rotated	76.6	61.6
Un-rotated	76.6	84.3
Un-rotated + Rotated	79.5	86.5
Time	73.7	81.6
Frequency	73.3	83.9
Frequency + Time	79.5	86.5

Table 5

The number of features kept and the accuracy of RF after each step of feature selection on both *itest* and *ivalid* set.

Features		Accuracy [%]	
		ivalid	itest
All features	1,696	86.9	82.6
Correlation removed	762	87.6	84.8
Wrapper	203	89.5	86.9
Wrapper strict	65	89.8	87.5

Table 6

The number of features selected from each sensor's data stream for the "Wrapper strict" feature subset.

Sensor	Acc	Gyro	Pressure	Mag	Ori
#Selected	30	15	1	7	11

In the next step, we more thoroughly investigate the impact of different feature types, by choosing different subsets of acceleration and magnetic field features. The subsets were made by either splitting the features based on whether they are un-rotated (Section 4.2), or based on whether they are computed in the time or frequency domain. The results are shown in Table 4. The most significant observation, is that no feature subset outperformed the whole set, confirming the usefulness of each feature type. It is worth pointing out that the features in the normalized coordinate system – which are not normally used in AR – consistently improved the accuracy by 2% (while not being good on their own, as there were simply fewer of them). Another interesting observation is that the frequency-domain accelerometer features outperformed the time-domain accelerometer features, even though the latter are more common in AR.

Finally, we present the effect of the feature selection method described in Section 4.4. The correlation threshold was set to 0.8. We tested two different improvement thresholds for the Wrapper step, resulting in "Wrapper" (improvement threshold set to 0.2%) and "Wrapper strict" (improvement threshold set to 0.05%). The results are listed in Table 5.

For the subsequent experiments we continued with the feature subset "Wrapper strict". Table 6 presents more information about this feature subset. From the results, it can be seen that most of the selected features are calculated from the acceleration data; followed by angular velocity and orientation data; next is the magnetometer data; and finally, only one feature is calculated from the pressure data.

8.3. Base models

After choosing the "Wrapper strict" feature set, we proceeded to test different ML algorithms. The results are presented in Table 7. For the classical ML models, the column "Default" presents the models' accuracy with the default scikit-learn parameters, and the column "Tuned" presents the models' accuracy after the 2-fold randomized parameter search. This search was repeated 10 times using different folds, averaging the results. For the end-to-end DL models, the column "Default" presents the models' performance with all sensors as input, and

Table 7

Accuracy achieved using different classical ML models and end-to-end DL models (including our end-to-end Multi-ResNet model) on the *itest* data. Standard deviation was calculated across ten repetitions of the experiment.

	<i>itest</i> Accuracy [%]		
	Default	Tuned	Std
SVM	81.1	84.1	0.8
DT	72.3	75.4	1.3
NB	76.7	76.7	0
KNN	79.3	79.3	0.2
MLP	82.0	86.7	0.4
RF	87.5	87.7	0.1
Bagging	83.4	87.1	0.4
GradBoost	88.6	89.4	0.3
AdaBoost	46.1	89.1	0.4
XGBoost	88.4	89	0.3
LabelSwitch	82.1	86.8	0.1
CNN-4	60.3	64.5	0.3
CNN-8	58.9	63.1	0.3
CNN-16	58.1	60.2	0.3
LSTM-2	60.1	62.3	0.2
CNN-4-LSTM-2	74.1	76.4	0.4
DNN-Spectrogram	80.1	81.2	0.4
Multi-ResNet-2	80.6	85.4	0.3
Multi-ResNet-4	84.8	89.2	0.2
Multi-ResNet-6	85.2	89.4	0.2

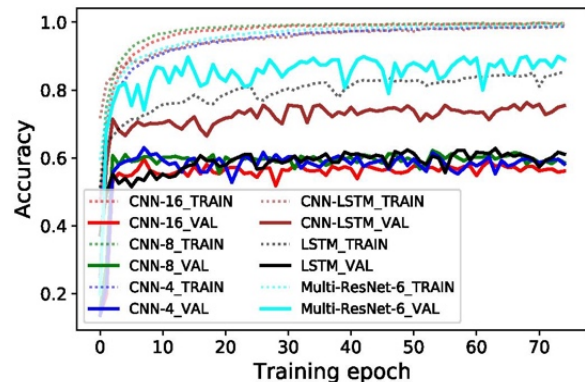


Fig. 11. Learning curves for the end-to-end DL models. The training accuracy (*itrain* data) is presented with dashed lines and the accuracy on the *ivalid* data is presented with solid lines.

the column "Tuned" presents the models' performance with the top three modalities as input (accelerometer, gyroscope and magnetometer). Multi-ResNet-2 is our DL method with 2 residual blocks per channel, Multi-ResNet-4 is the same method with 4 residual blocks per channel and so on. From the results, it can be seen that most of the classical ensemble models (i.e., RF, Bagging, Gradient Boosting, XGBoosting) performed better compared to the classical single models (i.e., SVM, DT, KNN and MLP). The highest accuracy of 89.4% is achieved by the Gradient Boosting (GradBoost) algorithm. The accuracy of the baseline DL methods is below 77%, which indicates that simple DL methods perform similar to the simple classical ML methods (e.g., DT and NB). In contrast, the accuracy of our DL method (Multi-ResNet) on the same *itest* set is 89.4%, which is basically the same as the best performing classical ML method.

In Fig. 11, we present the learning curves (accuracy with respect to the number of training epoch) for the end-to-end DL models. It can be seen that all of the models except the LSTM achieved training accuracy close to 100%. This indicates that the models converged with respect to the learning phase. However, for all of the baseline models the accu-

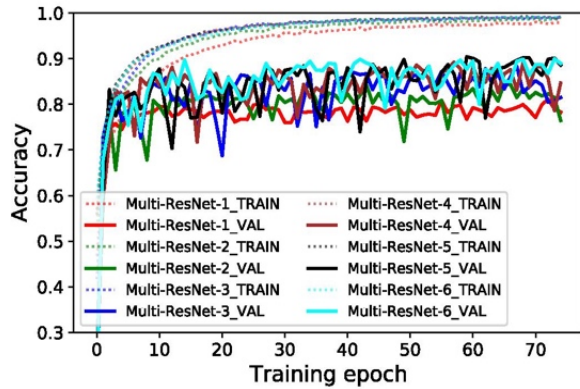


Fig. 12. Learning curves for the Multi-ResNet model with a varying number of residual blocks (from 1 to 6). The training accuracy (*itrain* data) is presented with dashed lines and the accuracy on the *ivalid* data is presented with solid lines.

Table 8

Accuracy achieved using different meta-models on the *itest* set. Standard deviation was calculated across the ten different data splits.

	<i>itest</i> Accuracy [%]	
	Mean	Std
meta-SVM	92.6	0.7
meta-DT	88.0	0.9
meta-NB	91.0	0
meta-KNN	92.1	0.3
meta-RF	93.0	0.1
meta-Bagging	93.0	0.4
meta-GradBoost	93.3	0.3
meta-AdaBoost	93	0.2
meta-XGBoost	93.4	0.3

racy on the *ivalid* data is close to 60%, except for the CNN-LSTM, which achieved accuracy of 75%. This indicates that simple architectures cannot achieve high accuracy. In contrast, our Multi-Resnet outperformed all of the baseline DL models with a margin of at least 10 percentage points.

Finally, in Fig. 12 we present learning curves for the Multi-ResNet with a varying number of residual blocks (from one to six blocks per channel). From the figure it can be seen that the accuracy increases as the number of residual blocks increases. This trend is present up to four residual block per channel. The Multi-ResNet-4, Multi-ResNet-5 and Multi-ResNet-6 perform similarly.

8.4. Ensemble of models

All the resulting models from the previous experiment (Table 7) were used as base models in an ensemble. Different meta-models were created using different ML algorithms and were tuned using a 2-fold randomized parameter search. This search was again repeated 10 times using different folds. From the results presented in Table 8, it can be seen that most meta-models (with a few exceptions) have similar accuracy, with the meta-model built with the XGB algorithm having the highest accuracy – 93.4%.

With the meta-XGB being the best meta-model, we analyzed which base models contribute to its performance. In particular, we were interested to find if both ML and DL models were required for the best results. To do so, we tried using only ML models, only DL models (in this case the default and tuned version of the Multi-ResNet), their combination, and only the best model from each category. The results are shown in

Table 9

Accuracy achieved using either (or both) classical ML or DL classifiers as the base of the ensemble on the *itest* set. Either all models of a type were used, or only the best one. Standard deviation was calculated across the ten different data splits.

Models used	<i>itest</i> Accuracy [%]	
	Mean	Std
All ML	91.3	0.3
All DL	89.8	0.3
All ML + All DL	93.4	0.3
Best ML + Best DL	93.8	0.2

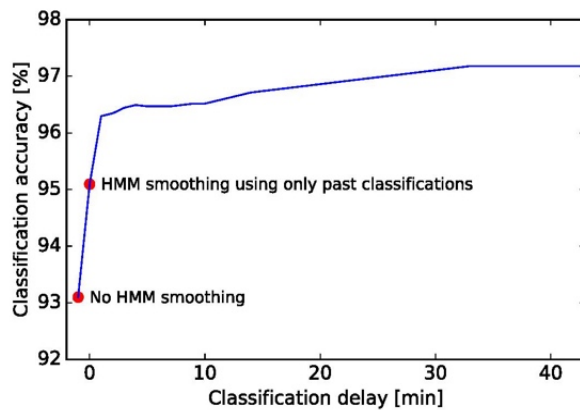


Fig. 13. Accuracy [%] for Markov smoothing applied with different delay intervals on the *itest* data.

Table 10

Different methods for smoothing the data. The test setting with no delay was used for comparison.

No smoothing	HMM	RNN	LSTM	Bi LSTM	GRU
93.4%	95.3%	93.7%	94.2%	93.5%	94.4%

Table 9, and show that having more than one ML model barely improves the performance; however, by combining the ML and DL models (which are substantially different), the accuracy increases significantly.

8.5. Markov smoothing

The last step was applying the HMM smoothing. Fig. 13 presents the accuracy with Markov smoothing applied using different delay intervals. The figure shows that even when working with no delay, using only past classifications to “smooth” the current one, the accuracy increased from 93.1% to 95.1%. Small delays slowly increased the accuracy up to its final value of 97.2%. By testing HMM smoothing with predictions made by different classifiers and ensembles, we noticed that the most substantial accuracy gains happen with 5–10 minute delays. Since the SHL Challenge allowed for an offline classification, we used the longest possible delay (using all of the data) for the challenge submission, as this was expected to yield the highest accuracy.

As an alternative to the HMM smoothing, we tested RNN, LSTM, bidirectional LSTM [62] and GRU [63] neural networks to smooth the predictions. These models used past and current predictions as the input and output the “corrected” current prediction. Results using these models are shown in Table 10. They all improved the prediction accuracy, but not to the extent of the HMM model.

Table 11

Accuracy [%] achieved after different steps of the proposed methodology. Test were performed on both internal and final test sets. Unless stated otherwise, each following step uses parameters of the previous one (e.g. all steps after the feature selection step, use the selected features). WL - window length. “/” denotes that ML and DL use different parameters.

	itest		SHL Test	
	ML	DL	ML	DL
WL - 5 sec, RF and frequency features for ML	77.1	78.7	85.1	78.8
WL - 60/40 sec	84.7	85.2	88.2	81.2
Using all features	82.6	/	89.8	/
Feature selection / Sensor stream selection	87.8	89.3	88.4	90.8
Using best classifier	89.4	/	90.9	/
Ensemble (Best ML + Best DL)	93.8		94.7	
Using HMM smoothing	97.2		96.0	

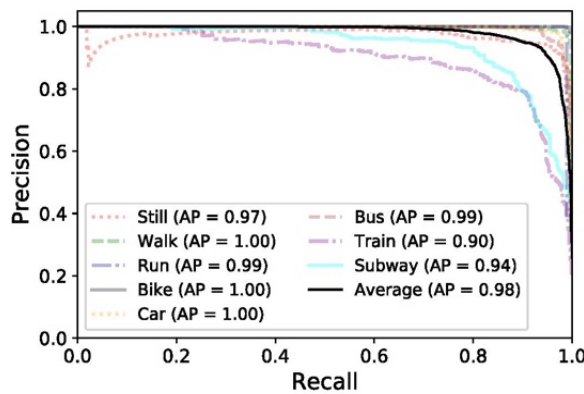


Fig. 14. Precision-recall curve for each of the activities and their average. The AP stands for the average precision, defined as $\sum_n (R_n - R_{n-1})P_n$, where R_n and P_n are the recall and precision for the n -th decision threshold.

8.6. Experiments on the SHL test set

By the time of writing this paper, the SHL challenge organizers had released the labels for the SHL Test set, thus enabling us to evaluate our methodology on it. Table 11 shows the different steps of our methodology on both the *itest* and the *SHL Test* set. For the *SHL Test*, set we used the same parameters as for the *itest* set, with no additional fitting to it. This table can serve as a summary on how much each step improves the performance. On the *itest* set, every step increased the performance as expected, with the biggest improvements being attributed to the window size (7 percentage points), feature selection (5 percentage points), ensemble (4 percentage points) and the use of HMM smoothing (4 percentage points). In the case of the *SHL Test* set, the starting accuracy was much higher (probably due to different activity distribution), and the role of feature selection was diminished – in fact, using all the features worked best. The sensor stream selection, ensemble and HMM smoothing provided similar benefits. The final results for both test sets are roughly the same.

Table 12 presents the confusion matrix and the precision, recall and F1 score for each class, achieved by this method on the *itest* data. From the confusion matrix, it can be seen that the most problematic activities are the *Train* and *Subway*, i.e., the only two activities that have an F1 score lower than 90%. This is not surprising, since the *Train* and *Subway* are in fact very similar. This also explains why most classifiers worked better on the *SHL Test* set than on the *itest* set, as the former contained proportionally far fewer of these two activities. An alternative display of the model's performance using Precision-Recall curves is presented in Fig. 14. Such curves would be useful to modify the decision threshold

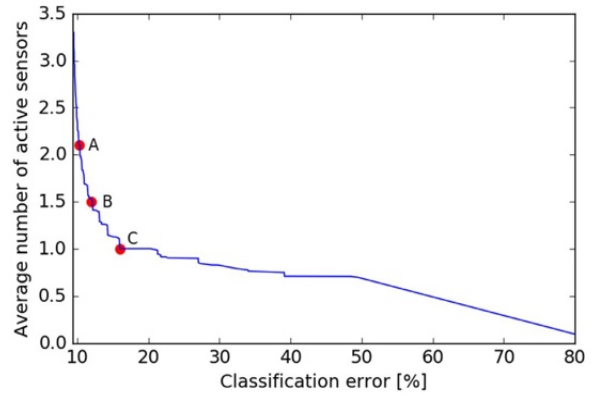


Fig. 15. Different trade-offs between the average number of active sensors and the classification error. Some sample solutions are marked with letters A, B and C.

if we would be particularly interested in correctly detecting a specific activity.

Table 13 presents the final results on the *SHL Test* and compares them to the results of other competitors. To be able to compare these results, we list the results using F1 score, as was used in the competition. The JSI-Deep and JSI-Classic are our methods which were ranked in first and second places at the SHL challenge, and have served as the basis for the development of the method presented in this paper, the meta-XGB-HMM. The last three rows are the third, fourth and the fifth ranked methods at the SHL challenge. From the results, it can be seen that our methods achieved an F1 score that is at least 3.6 percentage points higher, compared to the best performing method from the related work [64]. The meta-XGB-HMM presented in this paper yielded the highest F1 score of 94.9%, which is an increase of 6.1 percentage points compared to the best performing method from the related work.

8.7. Energy efficiency

To explore different trade-offs between energy efficiency and the classification error, we first calculated the Pareto front of the trade-offs that use different sensors for classifying different activities – as described in Section 7.1. The results are shown in Fig. 15. The most interesting solutions seem to lie in the area between one and two sensors used on average. On the same figure, three sample solutions are listed. Note that by not using the DL in the ensemble, the maximum achievable accuracy is 90%.

- Solution A: Accuracy lost: 0.5%, average number of sensors used: 2.1/5
This solution uses the accelerometer regardless of the activity. It uses the magnetometer in all vehicles and when standing still. In addition, it uses the barometer when walking and on the subway (perhaps to detect the pressure change that happens when the user enters or leaves a subway station). Somewhat peculiarly, it uses the orientation sensor when on the bus or train – potentially to use the rotated magnetometer features that can recognize the orientation of the magnetic field caused by the train.
- Solution B: Accuracy lost: 2%, average number of sensors used: 1.4/5
The accelerometer is used in all cases. The magnetometer is used when on the subway or train – the two vehicles that are hardest to distinguish from one another. All other sensors are inactive (with the exception of the gyroscope when running).
- Solution C: Accuracy lost: 7%, average number of sensors used: 1/5
Only the accelerometer is used in all cases.

Table 12

Confusion matrix, precision, recall and F1 score on the *itest* data for the model with highest accuracy, the meta-XGB-HMM model.

	Still	Walk	Run	Bike	Car	Bus	Train	Subway
Still	921	7	0	3	3	11	12	5
Walk	4	718	4	2	1	0	1	1
Run	0	9	328	0	0	0	0	0
Bike	3	0	0	508	0	0	0	0
Car	3	0	1	0	1272	0	0	0
Bus	10	8	1	1	0	876	4	0
Train	30	2	0	0	7	20	514	74
Subway	1	2	0	0	0	0	0	331
Recall [%]	95.7	98.2	97.3	99.4	99.7	97.3	79.4	99.1
Precision [%]	94.8	96.3	98.2	98.8	99.1	96.6	96.8	80.5
F1 score [%]	95.2	97.2	97.8	99.1	99.4	97.0	87.3	88.9

Table 13

Evaluation on the SHL challenge test data. Comparison of F1 scores between our work and the top ranked teams at the challenge.

Name	Algorithm	F1 score
(Our) meta-XGB-HMM	ML + DL + HMM	94.9%
(Our) JSI-Deep [26]	ML + DL + HMM	93.9%
(Our) JSI-Classic [27]	ML (XGBoost)	92.4%
(Rel. work) Tesaguri [64]	DL (spectrogram CNN)	88.8%
(Rel. work) S304 [25]	DL (feature MLP) + HMM	87.5%
(Rel. work) Conf. Matrix [24]	ML (RF)	87.5%

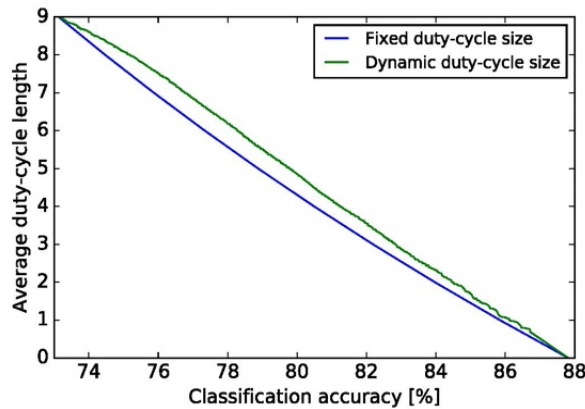


Fig. 16. Different trade-offs between the average duty-cycle length and the classification error on the *itest* data. Dynamic solutions are compared to those with a fixed duty-cycle length.

A quick take-away from the presented solutions is that if aiming to be energy efficient, we should always use the accelerometer and potentially use the magnetometer if we suspect the user is in a vehicle.

In the next step we took “Solution B” and applied the duty-cycle methodology on it (Section 7.2). Both fixed and activity-specific duty-cycle lengths were tried and plotted in Fig. 16. Duty-cycle length 0 means that the sensors are always active; duty cycle length 2 means that after a window of sensor readings (1 minute in our case), the sensors turn off for the duration of two windows. We can see that the Pareto front is mostly linear in both cases, decreasing the accuracy for roughly 1.6 percentage points (p.p.) for each additional increment of the average duty-cycle length by 1. However, for the same average duty-cycle length, the activity-specific assignment achieves roughly 1.5 p.p. higher accuracy in all cases.

In all the solutions, the same pattern appears: when in a vehicle, long duty-cycles are used, as these activities have infrequent transitions. When in one of the “hub” activities – *Still*, *Walk* – a short cycle is used

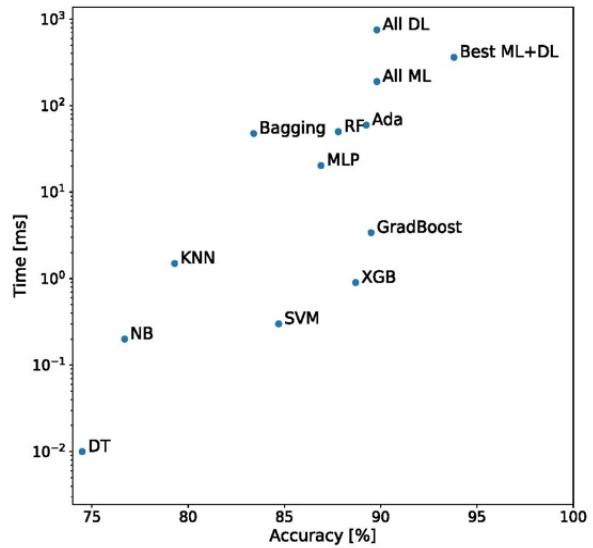


Fig. 17. Time for the classification of a single instance (note the logarithmic scale) for different methods. The “all DL”, “all ML” and “best ML + DL” represent the corresponding ensembles.

Table 14

A sample assignment of duty-cycle lengths to each activity.

Activity	Still	Walk	Run	Bike	Car	Bus	Train	Subway
Duty-cycle	2	1	0	1	4	3	2	3

for the opposite reasons. When running, a short duty-cycle is used due to this activity being short on average. An example solution with an average duty-cycle length of 2 is shown in Table 14.

Taking it all together, using the “Solution B”, the average duty-cycle length of 2 (Table 14) and the sampling frequency of 50 Hz, we lose roughly 5% (going from 90% to 85%) of the classification accuracy in exchange for using only 5% of all provided data.

As the additional post-processing step, one can use the HMM smoothing on the predictions generated by this solution. Doing so brings the accuracy to 89.9%, roughly 7 p.p. less than the best solution (97.2%) that uses the HMM smoothing.

Another way to minimize energy consumption would be to consider the energy spent on classifying the instances. This energy is largely proportional to the time spent on the task. We thus measured the time required for the classification of a single instance for different methods (the cost of the HMM is independent of the method and not included). All the measurements were performed on a desktop computer, but it is

reasonable to assume that the results on a smartphone would be roughly proportional. The deep-learning methods were tested of the CPU (and not the GPU) as related work showed that the GPU does not greatly accelerate the classification time on a smartphone [65].

The results in Fig. 17 show an exponential relationship between the classification accuracy and the required processing time. The XGBoost and GradBoost algorithms seem to present the best trade-offs between the performance and the energy consumption, losing roughly 4 p.p. of accuracy compared to the ensemble, which is three orders of magnitudes slower.

9. Conclusion and discussion

The SHL Dataset is a uniquely large and sensor-rich dataset collected in a real-life setting. It provides an open platform for creating and testing algorithms for AR and similar tasks. By containing activities not common in AR datasets, such as *Train* and *Subway*, it opens new challenges for the AR community. The SHL Challenge was an effective way to jumpstart the research on the dataset, yielding the first solutions to the basic locomotion AR problem.

This paper offers a detailed description of our method, which won the SHL challenge by most accurately predicting the activities on an unlabelled dataset. It can thus be considered the reference AR method for the SHL dataset, and a good starting point for similar AR problems. The highlights of the method are the complex pipeline that includes novel pre-processing steps (such as coordinate system rotation), the comprehensive feature set, the complex feature selection method and the novel neural network architecture for deep learning. We also presented how to optimize the energy consumption of our method by adapting sensor settings to each activity. Their key contribution, however, might be the extensiveness of the performed experiments, which give insights on the effectiveness of different methods on the SHL dataset and in similar domains.

The following are our key experimental observations: (**Pre-processing**) The ordering of the data before making the split for internal evaluation was a key step, as otherwise the results on the *itest* set would be too good due to overfitting, and would not translate to the *SHL Test* set. (**Window size**) The window size used for the segmentation influences both classical and DL methods similarly, i.e., a higher accuracy is achieved with longer windows. (**Sampling frequency**) The features extracted for the classical ML are less sensitive to the data sampling frequency compared to the automatic features learned by the Multi-ResNet. In both classical ML and DL cases, a frequency of at least 20 Hz was required for most accurate predictions, but surprisingly good results were achieved even with much lower frequencies (e.g. 1 Hz). (**Sensors**) Comparing sensors individually, the accelerometer data provided the highest accuracy for both the classical ML and DL methods. The accelerometer and magnetometer proved to be the best pair. However, all sensors had to be fused together to achieve the best results. (**Feature subgroups**) For the classical ML, using features either in time or frequency domains yielded roughly the same results. Using both feature groups improved the accuracy. The accuracy was further increased by adding features from the accelerometer and magnetometer that were rotated into the world coordinate system. (**Feature selection**) The feature selection results are in line with the per-sensor results - i.e., more features were kept from the sensors for which the sensor-specific models achieved higher accuracy. The three-step feature selection method found a good feature subset that did not overfit to the *valid* set. This increased the accuracy by roughly 5 p.p. on the *itest* set. Surprisingly, on the *SHL Test* set, using all the features worked better than any tested feature subset. (**Classification models**) As expected, ensemble models (e.g., boosting and RF) performed better than single models (e.g., KNN and DT). The Multi-ResNet, using an end-to-end DL architecture, achieved an accuracy of 89.4%, on a par with the classical ML models - the best of which (GradBoost) also achieved an accuracy of 89.4%. Considering that DL has enjoyed limited success in AR so far, it may be that the research

community is still developing appropriate architectures. The majority of the models got significantly better after the hyper-parameter tuning. (**Ensembles**) Building an ensemble that combined the ML and DL models significantly improved the accuracy (from 89.4% to 93.8%). (**HMM**) The HMM smoothing also worked surprisingly well, improving the accuracy by four percentage points (from 93.8% to 97.2%). Thus, for the type of data where the activities are reasonably long-lasting (in the SHL challenge all activities last more than 12 minutes on average - Table 1), we recommend the use of this post-processing step. (**Energy efficiency**) By combining three different approaches for optimizing the energy-consumption, we found a solution that reduces the amount of data needed - and thus the expected energy to collect and process it - by 95%, while losing only 5% of accuracy.

Regarding the limitations of the presented work, the windowing and frequency experiments for the Multi-ResNet are biased towards the specific DL architecture. Different architectures might find other parameters more suitable. Also, the final and best performing feature subset ("Wrapper strict") might be biased towards RF, since RF was used to build models for the wrapper feature-selection method. Finally, the SHL challenge data comes from a single smartphone, worn by the same user in his trouser pocket for a period of four months. Thus, the presented analysis is person dependent and the models are person-specific.

Parts of this methodology were successfully used in the past on other datasets (the deep learning component [66,67] and the classical machine learning component [6,34]), which can be an indicator of the generality of the proposed approach. Additionally, we used very similar methodology to win the 2019 version of the SHL competition [68] (this paper was about the competition in 2018). However, in the future, we plan to evaluate the models on additional subjects as the data becomes available, and on additional datasets for AR. Additionally, the Multi-ResNet can be updated with more advanced end-to-end fusion approaches like two-stream network fusion [69] or multimodal subspace clustering [70].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRedit authorship contribution statement

Martin Gjoreski: Conceptualization, Formal analysis, Investigation, Methodology, Validation, Visualization, Software, Writing - review & editing. **Vito Janko:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Visualization, Software, Writing - review & editing. **Gašper Slapničar:** Formal analysis, Data curation, Software. **Miha Mlakar:** Formal analysis, Data curation, Software. **Nina Reščič:** Formal analysis, Data curation, Software. **Jani Bizjak:** Formal analysis, Data curation, Software. **Vid Drobnič:** Formal analysis, Data curation, Software. **Matej Marinko:** Formal analysis, Data curation, Software. **Nejc Mlakar:** Formal analysis, Data curation, Software. **Mitja Luštrek:** Conceptualization, Supervision, Writing - review & editing, Project administration, Funding acquisition. **Matjaž Gams:** Conceptualization, Supervision, Writing - review & editing, Project administration, Funding acquisition.

Acknowledgments

The authors acknowledge the financial support from the Slovenian Research Agency (Grant U2-AG-16/0672, 0287 and grant U2-RP-19/0441, 0209).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.inffus.2020.04.004.

References

- [1] Happiness and Satisfaction with Work Commute, (<https://link.springer.com/article/10.1007/s11205-012-0003-2>).
- [2] H. Gjoreski, M. Ciliberto, L. Wang, F.J. Ordonez Morales, S. Mekki, S. Valentin, D. Roggen, The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices, *IEEE Access* (2018).
- [3] S. Kozina, H. Gjoreski, M. Gams, M. Luštrek, Efficient activity recognition and fall detection using accelerometers, in: *International Competition on Evaluating AAL Systems Through Competitive Benchmarking*, Springer, 2013, pp. 13–23.
- [4] D. Roggen, A. Calatroni, M. Rossi, T. Holleczeck, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirk, A. Ferscha, et al., Collecting complex activity datasets in highly rich networked sensor environments, in: *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, IEEE, 2010, pp. 233–240.
- [5] V. Janko, B. Cvetković, A. Gradišek, M. Luštrek, B. Štrumbelj, T. Kajtna, E-gibalec: mobile application to monitor and encourage physical activity in schoolchildren, *J. Ambient Intell. Smart Environ.* 9 (5) (2017) 595–609.
- [6] B. Cvetković, R. Szelicki, V. Janko, P. Lutowski, M. Luštrek, Real-time activity monitoring with a wristband and a smartphone, *Inf. Fusion* 43 (2018) 77–93.
- [7] H. Guo, L. Chen, L. Peng, G. Chen, Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble, in: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ACM, 2016, pp. 1112–1123.
- [8] P. Zappi, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, G. Troster, Activity recognition from on-body sensors by classifier fusion: sensor scalability and robustness, in: *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, IEEE*, 2007, pp. 281–286.
- [9] H. Gjoreski, J. Bizjak, M. Gjoreski, M. Gams, Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer, in: *Proceedings of the IJCAI 2016 Workshop on Deep Learning for Artificial Intelligence*, New York, NY, USA, 10, 2016.
- [10] D. Ravi, C. Wong, B. Lo, G.-Z. Yang, A deep learning approach to on-node sensor data analytics for mobile or wearable devices, *IEEE J. Biomed. Health Inform.* 21 (1) (2017) 56–64.
- [11] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, *Proc. IEEE* 86 (11) (1998) 2278–2324.
- [12] S. Bhattacharya, N.D. Lane, Sparsification and separation of deep learning layers for constrained resource inference on wearables, in: *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, ACM, 2016, pp. 176–189.
- [13] M. Zeng, L.T. Nguyen, B. Yu, O.J. Mengshoel, J. Zhu, P. Wu, J. Zhang, Convolutional neural networks for human activity recognition using mobile sensors, in: *6th International Conference on Mobile Computing, Applications and Services, IEEE*, 2014, pp. 197–205.
- [14] J. Yang, M.N. Nguyen, P.P. San, X.L. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, in: *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [15] F. Ordóñez, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, *Sensors* 16 (1) (2016) 115.
- [16] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
- [17] K. Krishna, D. Jain, S.V. Mehta, S. Choudhary, An lstm based system for prediction of human activities with durations, *Proc. ACM Interact. Mob. Wearable Ubiquit. Technol.* 1 (4) (2018) 147.
- [18] V.S. Murahari, T. Plötz, On attention models for human activity recognition, in: *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, ACM, 2018, pp. 100–103.
- [19] B. Martin, V. Addona, J. Wolfson, G. Adomavicius, Y. Fan, Methods for real-time prediction of the mode of travel using smartphone-based gps and accelerometer data, *Sensors* 17 (9) (2017), doi:10.3390/s17092058.
- [20] S.-H. Fang, H.-H. Liao, Y.-X. Fei, K.-H. Chen, J.-W. Huang, Y.-D. Lu, Y. Tsao, Transportation modes classification using sensors on smartphones, *Sensors* 16 (8) (2016) 1324.
- [21] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, M. Srivastava, Using mobile phones to determine transportation modes, *ACM Trans. Sensor Netw. (TOSN)* 6 (2) (2010) 13.
- [22] S. Hemminki, P. Nurmi, S. Tarkoma, Accelerometer-based transportation mode detection on smartphones, in: *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, in: *SenSys '13*, ACM, New York, NY, USA, 2013, pp. 131–134, doi:10.1145/2517351.2517367.
- [23] L. Wang, H. Gjoreski, K. Muraom, T. Okita, D. Roggen, Summary of the sussex-huawei locomotion-transportation recognition challenge, in: *Proceedings of the 6th International Workshop on Human Activity Sensing Corpus and Applications (HASCA2018)*, Springer, 2018, pp. 1521–1530.
- [24] A.D. Antar, M. Ahmed, M.S. Ishrak, M.A.R. Ahad, A comparative approach to classification of locomotion and transportation modes using smartphone sensor data, in: *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, in: *UbiComp '18*, ACM, New York, NY, USA, 2018, pp. 1497–1502, doi:10.1145/3267305.3267516.
- [25] P. Widhalm, M. Leodolter, N. Brändle, Top in the lab, flop in the field?: Evaluation of a sensor-based travel activity classifier with the shl dataset, in: *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, in: *UbiComp '18*, ACM, New York, NY, USA, 2018, pp. 1479–1487, doi:10.1145/3267305.3267514.
- [26] M. Gjoreski, V. Janko, N. Reščić, M. Mlakar, M. Luštrek, J. Bizjak, G. Slapničar, M. Marinko, V. Drobnič, M. Gams, Applying multiple knowledge to sussex-huawei locomotion challenge, in: *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, in: *UbiComp '18*, ACM, New York, NY, USA, 2018, pp. 1488–1496, doi:10.1145/3267305.3267515.
- [27] V. Janko, N. Reščić, M. Mlakar, V. Drobnič, M. Gams, G. Slapničar, M. Gjoreski, J. Bizjak, M. Marinko, M. Luštrek, A new frontier for activity recognition: The sussex-huawei locomotion challenge, in: *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, in: *UbiComp '18*, ACM, New York, NY, USA, 2018, pp. 1511–1520, doi:10.1145/3267305.3267518.
- [28] Sussex-Huawei Locomotion Challenge, (<http://www.shl-dataset.org/activity-recognition-challenge>).
- [29] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, D. Roggen, Benchmarking the shl recognition challenge with classical and deep-learning pipelines, in: *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, in: *UbiComp '18*, ACM, New York, NY, USA, 2018, pp. 1626–1635, doi:10.1145/3267305.3267531.
- [30] J. Lee, J. Kim, Energy-efficient real-time human activity recognition on smart mobile devices, *Mob. Inf. Syst.* 2016 (2016). Article ID 2316757, 12 pages.
- [31] B. Sefer, S. Baumbach, A. Dengel, S. Abdennadher, Human activity recognition, in: *Proceedings of the 8th International Conference on Agents and Artificial Intelligence*, SCITEPRESS-Science and Technology Publications, Lda, 2016, pp. 488–493.
- [32] J.B. Kuipers, et al., Quaternions and rotation sequences, 66, Princeton university press Princeton, 1999.
- [33] X. Su, H. Tong, P. Ji, Activity recognition with smartphone sensors, *Tsinghua Sci. Technol.* 19 (3) (2014) 235–249, doi:10.1109/TST.2014.6838194.
- [34] B. Cvetković, V. Janko, M. Luštrek, Demo abstract: Activity recognition and human energy expenditure estimation with a smartphone, in: *Pervasive Computing and Communication Workshops (PerCom Workshops)*, 2015 IEEE International Conference on, IEEE, 2015, pp. 193–195.
- [35] tsfresh, (<http://tsfresh.readthedocs.io/en/latest/>).
- [36] Mutual info score, (http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html).
- [37] Pearson correlation coefficient, (https://en.wikipedia.org/wiki/Pearson_correlation_coefficient).
- [38] scikit-learn, (<https://scikit-learn.org/stable/>).
- [39] T.K. Ho, Random decision forests, in: *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1, IEEE, 1995, pp. 278–282.
- [40] J.R. Quinlan, Improved use of continuous attributes in c4.5, *J. Artif. Intell. Res.* 4 (1996) 77–90.
- [41] S.J. Russell, P. Norvig, *Artificial intelligence: A modern approach*, Malaysia; Pearson Education Limited, 2016.
- [42] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, *Mach Learn* 6 (1) (1991) 37–66.
- [43] J. Shawe-Taylor, N. Cristianini, *An introduction to support vector machines and other kernel-based learning methods*, 204, Cambridge University Press Cambridge, 2000.
- [44] L. Breiman, Bagging predictors, *Mach. Learn.* 24 (2) (1996) 123–140.
- [45] F. Rosenblatt, Principles of neurodynamics. perceptrons and the theory of brain mechanisms, Technical Report, Cornell Aeronautical Laboratory, 1961.
- [46] G. Martínez-Muñoz, A. Sánchez-Martínez, D. Hernández-Lobato, A. Suárez, Class-switching neural network ensembles, *Neurocomputing* 71 (13–15) (2008) 2521–2528.
- [47] G. Martínez-Muñoz, A. Suárez, Switching class labels to generate classification ensembles, *Pattern Recognit.* 38 (10) (2005) 1483–1494.
- [48] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, *arXiv Preprint arXiv:1502.03167* (2015).
- [49] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.
- [50] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, 2016, pp. 770–778.
- [51] Z. Wang, W. Yan, T. Oates, Time series classification from scratch with deep neural networks: A strong baseline, in: *2017 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2017, pp. 1578–1585.
- [52] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, *IEEE Press*, 2001.
- [53] hmmlearn, (<https://hmmlearn.readthedocs.io/en/latest/>).
- [54] V. Janko, M. Luštrek, Energy-efficient data collection for context recognition, in: *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, ACM, 2017, pp. 458–463.
- [55] V. Janko, M. Luštrek, Choosing duty-cycle parameters for context recognition, in: *2018 14th International Conference on Intelligent Environments (IE)*, IEEE, 2018, pp. 83–86.
- [56] A. Carroll, G. Heiser, et al., An analysis of power consumption in a smartphone., in: *USENIX Annual Technical Conference*, 14, Boston, MA, 2010, p. 21.
- [57] I. Crk, F. Albinati, C. Gniady, J. Hartman, Understanding energy consumption of sensor enabled applications on mobile phones, in: *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, IEEE, 2009, pp. 6885–6888.
- [58] N.D. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi, F. Kawsar, Squeezing deep learning into mobile and embedded devices, *IEEE Pervasive Comput.* 16 (3) (2017) 82–88.

- [59] M. Tan, Q.V. Le, Efficientnet: rethinking model scaling for convolutional neural networks, arXiv Preprint arXiv:1905.11946 (2019).
- [60] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: nsga-ii, *IEEE Trans. Evol. Comput.* 6 (2) (2002) 182–197.
- [61] A. Khan, N. Hammerla, S. Mellor, T. Plötz, Optimising sampling rates for accelerometer-based human activity recognition, *Pattern Recognit. Lett.* 73 (2016) 33–40.
- [62] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, *IEEE Trans. Signal Process.* 45 (11) (1997) 2673–2681.
- [63] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).
- [64] C. Ito, X. Cao, M. Shuzo, E. Maeda, Application of cnn for human activity recognition with fft spectrogram of acceleration and gyro sensors, in: *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, in: *UbiComp '18*, ACM, New York, NY, USA, 2018, pp. 1503–1510, doi:10.1145/3267305.3267517.
- [65] S. Richoz, A. Perez-Urbe, P. Birch, D. Roggen, Benchmarking deep classifiers on mobile devices for vision-based transportation recognition, in: *Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ACM, 2019, pp. 803–807.
- [66] M. Gjoreski, S. Kalabakov, M. Luštrek, H. Gjoreski, Cross-dataset deep transfer learning for activity recognition, in: *Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ACM, 2019, pp. 714–718.
- [67] G. Slapničar, N. Mlakar, M. Luštrek, Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network, *Sensors* 19 (15) (2019) 3420.
- [68] L. Wang, H. Gjoreski, M. Ciliberto, P. Lago, K. Murao, T. Okita, D. Roggen, Summary of the sussex-huawei locomotion-transportation recognition challenge 2019, in: *Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ACM, 2019, pp. 849–856.
- [69] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.
- [70] M. Abavisani, V.M. Patel, Deep multimodal subspace clustering networks, *IEEE J. Sel. Top. Signal Process.* 12 (6) (2018) 1601–1614.

5.4 Summary and Discussion

This chapter presented three studies in which the proposed general method (depicted in Figure 3.1) that combines expert knowledge, classical ML, and DL was applied. In the three publications, the method’s architecture was also presented. The method, presented in Figure 3.1, improved the performance of the baseline models for extracting human physical, physiological and psychological information from wearable sensor data.

Table 5.1 presents a summary of the general method’s components that were used in the studies. Differently from the early-work studies (summarized in Table 4.9), end-to-end DL was used in these studies. More specifically, the proposed STResNet was compared to a variety of state-of-the-art DL methods in the studies on driving distractions and locomotion recognition. In both of these studies, STResNet achieved the best performance. Table 5.2 presents a summary of the experimental results.

Similar to that discussion, the experimental results from the different studies can be compared using the study-specific baseline methods as a comparison proxy. The column “Combining modalities” summarizes the results for different modalities. For the CHF study, only one modality was available (sound), and thus combining modalities was not plausible. For the driving distractions study, the model based on a single modality, XGB using facial activation units as input, achieved the best results. However, the models built using all modalities or using feature selection achieved similar results. For the locomotion recognition study, the results from the best single modality were improved using a combination of several modalities and feature selection. Consequently, **the first hypothesis**, i.e., combining data from a variety of wearable sensors improves the performance, **was confirmed** in one of the two studies in which we were able to test it.

Table 5.1: Method summary. Presenting which components of the proposed general method (Figure 3.1) were used in the studies that combined ML and DL approaches.

Method’s component	Study		
	CHF	Driving distractions	Locomotion
Sensors	Sound (phonocardiogram)	Physiological + video-based	Smartphone
Dataset	Ours + 5 public datasets	Public dataset	Public dataset
Experts	ML + Medical	ML	ML
Preprocessing	All	All	All
Feature Extraction	All	All	All
Feature Selection	Ranking	Ranking	Wrapper + ranking
ML	RF	Many (classification)	Many (classification)
DL	STResNet	Many (+ STResNet)	Many (+ STResNet)
Fusion	Meta (ML + DL)	Meta (ML + DL)	Meta (ML + DL)

The column “Combining methods or datasets” summarizes the results for meta-learning vs. no meta-learning. Across all three studies (CHF, driving distractions, and locomotion recognition) the overall results were improved using a meta-learner that combined

predictions from classical (feature-based) models and the end-to-end DL model (STResNet). More specifically, in the study on CHF monitoring, the baseline results were improved using a meta-learner that combined segment-specific predictions from the Random Forest model and the STResNet model. In the study on locomotion recognition, the baseline results were improved using a meta-learner that combined the predictions from the XGB model and the STResNet model. In addition to that, the meta-learner modeled temporal dependencies between different locomotion activities, which was done using algorithms that are capable of modeling such regularities (HMMs and LSTMs).

Finally, in the study on monitoring driving distractions, the meta-learning analysis was performed after the publication of the paper. Similarly to the previous two studies, the predictions of the best-performing classical model (XGB) and the predictions of the STResNet model were used as input to a meta-learner, which was then evaluated using a LOSO evaluation technique. The meta-learner improved the results from an F1-score of 78%, achieved by the best-performing single model (XGB), to an F1-score of 82%.

Table 5.2: Results summary. The numbers in the table represent the evaluation score used in each study (Accuracy or F1-score).

Study (metric)	Method	Combining modalities	Combining methods or datasets
CHF (Accuracy)	Baseline	86.7, (RF, CHF paper – Table 6)	86.7 (RF, CHF paper – Table 6)
	Comb.	/	92.9 (RF + STResNet, same table)
Driving Distr. (F1-score)	Baseline	78 (XGB with AUs Driving distr. paper – Table 7)	78 (XGB with AUs, Driving distr. paper – Table 7)
	Comb.	77 (XGB with selected features, same table)	82 (XGB + STResNet, after publishing)
Locomotion (F1-score)	Baseline	86.5 (RF with acc. data, Locom. paper – Table 3)	89.4 (GradBoost, Locom. paper – Table 7)
	Comb.	89.4 (GradBoost and Multi-ResNet-6, Locom. paper – Table 7)	93.8 (GradBoost + STResNet, Locom. paper – Table 9)

By examining the overall results in Table 5.2, it can be concluded that the meta-learners improved the baseline results in all the studies, which experimentally **confirms the second hypothesis of the thesis**: “Our method for combining machine-learning models, including end-to-end deep learning, enables additional meta-learning layer that can combine models to maximize performance, can learn personalized models and can learn models that take into account the temporal dependence in the data, which improves the performance even further for monitoring human physical, physiological and psychological states”.

Chapter 6

Conclusions

This thesis addressed the problem of combining expert knowledge, classical ML, and DL to enable the transformation of wearable sensor data into valuable health and behavior information. The method uses end-to-end learning on sensor data as a standalone approach or, in combination with classical machine learning, to produce beyond state-of-the-art performance. The method is presented formally through definitions, algorithms, and flowcharts. The method is general and can be applied to many mobile health and behavior-monitoring problems for which wearable sensors are available.

To examine the validity of the thesis hypotheses, we applied the proposed method in seven health and behavior-monitoring domains: chronic heart failure detection from heart sounds, locomotion recognition from smartphone sensors, the detection of driving distractions from physiological and video-based sensors, stress, emotion and cognitive-load recognition from physiological sensors, and blood pressure estimation from ECG sensors.

The **first hypothesis** of the thesis is: “Combining data from a variety of wearable sensors improves the performance for monitoring human physical, physiological and psychological states using machine learning”. This hypothesis was experimentally **confirmed for four of the seven experimental domains**. In two domains, the combination of sensor data was not possible because only one type of sensor data was available and in one domain, the results were equally good both with and without combined modalities. In the rest of the experimental domains, the combination of different wearable sensors always outperformed the single-sensor approaches. One reason for this is that the wearable sensor data is noisy. Thus, by enabling several sources of information and several learners, the impact of the noise on the performance is lowered.

Additionally, the different sensors enable different types of information. For example, heart-related sensors and HRV analysis provide information both for the activation of the sympathetic and of the parasympathetic nervous system. In contrast, the GSR sensor provides information only for the activation of the sympathetic nervous system. Thus, by combining these two types of sensors, better modeling of the participants’ psychological, physiological and behavioral states can be achieved.

The **second hypothesis** of the thesis is: “Our method for combining machine-learning models, including end-to-end deep learning, enables an additional meta-learning layer that can combine models to maximize performance, can learn personalized models and can learn models that take into account the temporal dependence in the data, which improves the performance even further for monitoring human physical, physiological and psychological states”. **This hypothesis was also confirmed experimentally. Across all seven experimental domains**, the baseline results were improved using meta-learning. In the stress study, the meta-learner combined predictions from several classifiers. In the blood-

pressure study, the meta-learner performed person-specific calibration. In the emotion-recognition study, the meta-learner utilized knowledge from six different datasets. In the cognitive-load study, a multi-task learning method combined two datasets. In the studies on CHF monitoring, locomotion recognition and driver distraction monitoring, the baseline results were improved using a meta-learner that combined the predictions from a classical ML classifier STResNet. This is because STResNet learns directly from the sensor data; thus, it has the potential to discover patterns in the data, not present in the expert-defined features.

Finally, Table 6.1 summarizes the seven studies in relation to the two thesis hypotheses. The colored (green) cells show which studies confirmed the hypotheses and the white cells show which studies did not confirm the hypotheses or were not tested because the combination of sensor data was not possible.

Table 6.1: Experimental domains and their relation to the thesis hypotheses.

Study	Hypothesis 1	Hypothesis 2
CHF	Not tested (only heart sounds)	A meta-learner that combines predictions of a classical and an end-to-end DL classifier
Locomotion	Combination of smartphone sensors + feature selection improved the results achieved by the best single modality (an independent source of data)	A meta-learner that combines predictions of a classical and an end-to-end DL classifier, and takes into account temporal dependencies
Driving	Facial Activation Units were equally as good as the combination from several modalities	A meta-learner that combines predictions of a classical and an end-to-end DL classifier
Stress	Combination of physiological sensors improved the results achieved by the best single modality	A meta-learner that combines predictions of several classical classifiers
Blood pressure	Not tested (only ECG sensor)	A personalized meta-learner that combines predictions of several classical classifiers
Emotions	Combination of physiological sensors improved the results achieved by the best single modality	A meta-learner that combines predictions of several classical classifiers
Cognitive load	Combination of physiological sensors improved the results achieved by the best single modality	Combining two datasets using multi-task learning

6.1 Summary of Contributions

This thesis generated the following original contributions:

1. **A novel general method that combines expert knowledge, classical machine learning, and deep learning for extracting human physical, physiological, and psychological information from wearable sensor data.** The method includes learning from a large-body of expert-defined features extracted from the processed data using classical ML, and end-to-end DL that learns directly from the sensor data with the potential to discover new useful patterns in the data, previously unknown to experts. An additional novelty of the method relies on the fusion of classical machine learning and end-to-end deep learning approaches, which combines models to maximize performance, can learn personalized models, and/or can learn models that consider the temporal dependence in the data. The method was presented formally through definitions, algorithms, and flowcharts.
2. **A novel deep learning architecture for end-to-end learning specialized for multimodal sensor data.** In contrast to the known end-to-end deep learning architectures, STResNet can learn from several sensors simultaneously, each of them having its own sampling frequency. Additionally, STResNet can learn both in the time domain and in the frequency domain, which is essential for health and behavior monitoring with wearable sensors. STResNet was successfully applied in three different domains, including activity recognition, monitoring driving distractions and chronic heart failure detection. Additionally, STResNet has been successfully applied in a fourth study on blood pressure estimation from BVP signals, in a study separate from this thesis [42]. Each of these studies had a variety of sensors, including smartphone sensors, sound and physiological sensors, which suggests that STResNet is a general end-to-end DL architecture.
3. **Unified application of the method on seven domains:** The method is general and can be applied to many mobile health and behavior-monitoring problems for which wearable sensors are available. This thesis presented the method's application on seven domains with various combinations of sensors.
4. **Collecting new datasets and creating publicly available software for mobile health and behavior monitoring with wearable sensors:** New labeled datasets were created in the studies on stress, cognitive load, Chronic Heart Failure and blood pressure. In addition to the datasets, the software includes components for feature extraction (the list of features that can be extracted is available in Appendix A), a component for learning models with STResNet, and a component for the fusion of classical and end-to-end deep learning that won an international machine-learning competition at the UbiComp conference.

6.2 Limitations and Future Directions

6.2.1 Threats to Validity

The thesis findings are conditioned by the experimental setups in which the studies were performed. For example, the study on driver distraction monitoring was performed on data collected in a simulated environment. Thus, the next step would be a real-life study in order to confirm the validity of the results from the study on driver distraction monitoring. Overfitting is another threat to validity for any ML-based and data-driven study. In this thesis, evaluation approaches, which are known to provide robust results, were used. For example, train-validation-test splits and leave-one-subject-out evaluations. In the study on locomotion recognition from smartphone sensors, all the data was collected by one subject and the models were person-specific. In order to confirm the validity of these models, the approach needs to be tested on data collected by additional participants.

6.2.2 From Supervised to Self-Supervised and Unsupervised Learning

This thesis focused on supervised learning. However, supervised learning is expensive as it requires labeled data. For that reason, a promising research direction is approaches that can decrease the requirement for labeled data by allowing the transfer of knowledge/models/data between domains. In this thesis, we performed early work on transfer learning and multi-task learning, but these fields should be analyzed in more detail. Additionally, few-shot learning and self-supervised learning [199] can be explored. Furthermore, approaches that allow unsupervised domain adaptation can be even more interesting as they completely eliminate the requirement for labeled data. For example, generative models could be one approach for unsupervised domain adaptation.

6.2.3 Explainable and Responsible AI

Explainable AI refers to methods and techniques in the application of AI such that the results of the solution can be understood by human experts. Explainable AI has the potential to minimize trade-offs between the transparency and the performance of AI models. More importantly, however, it would bring additional trust in AI solutions by users from different non-technical fields. For example, this thesis proposed a method for CHF monitoring that was developed in collaboration with medical doctors. This method would be much more easily adopted for wider usage if it offers explanations about different predictions made by the ML models. Providing explanations by ML models is also in line with the EU's General Data Protection Regulation ("GDPR"). According to the GDPR, the ML models should offer the possibility to answer/provide explanations such as: "You were denied a loan because your annual income was £30,000. If your income had been £45,000, you would have been offered a loan". This ML domain is called "searching for counterfactual explanations" [200][201][202].

Responsible AI is closely related to explainable AI; it is a methodology for the large-scale implementation of AI methods in real organizations with fairness, model explainability, and accountability at its core [203]. For example, non-ethical usage for the purpose of military or business¹⁵ gains must be avoided.

¹⁵ <https://www.businessinsider.com/china-emotional-surveillance-technology-2018-4>

6.2.4 Cost of AI Systems

This thesis has presented real-life use-cases of how AI-based methods can improve our lives. However, the widespread usage of AI-based products comes with a series of hidden costs. The essay “Anatomy of an AI System” [204] investigated the human labor, data, and planetary resources required to build and operate a state-of-the art AI product, such as Amazon Echo. The overall process includes: obtaining lithium from Bolivia, collecting labeled datasets in Southeast Asia, shipping and international logistics, analyzing data by AlexaVoiceService through distributed data centers, and electronic waste storage in Ghana, Pakistan, and China. The adverse effects of new technological and engineering systems need to be addressed, as the benefits may lower than the costs required to build and operate them. AI is powerful and has much potential. It is up to us to find the right paths to maximize its positive impact.

Appendix A

Features

Statistical Features

1. Mean
2. Standard deviation
3. Skewness
4. Kurtosis
5. 25th quartile
6. 75th quartile
7. Quartile deviation
8. Difference between max and min values
9. Coefficient of variation
10. Mean value of first derivative
11. Mean value of second derivative

HRV Features

1. Mean heart rate
2. Mean R-R interval
3. Standard deviation of R-Rs (sdnn)
4. Standard deviation of the differences between all subsequent R-R intervals (sdsd)
5. Root of the squared differences (rmssd)
6. Proportion of interval differences of successive R-Rs greater than 20 ms (pnn20)
7. Proportion of interval differences of successive R-Rs greater than 50 ms (pnn50)
8. Poincare plot SD1 (sd)
9. Poincare plot SD2 (sd2)
10. Poincare plot SD1/SD2 (sd1/sd)

Galvanic Skin Response Features (GSR Features)

1. Power of the raw signal

2. Number of skin conductance responses (SCR)
3. SCRs per second
4. Power of the SCRs
5. Sum of the signals positive derivative (segments where the signal increases)
6. Proportion between the positive derivative and the negative derivative
7. Mean value of the derivative of the tonic component (slow changing GSR component)
8. Mean value of the difference between the raw signal and the tonic component
9. Total spectral power of the signal in five frequency bands between 0 Hz and 0.6 Hz with a 0.1 Hz span (6 features)
10. Amplitude increase of the biggest SCR
11. Amplitude decrease of the biggest SCR
12. Biggest SCR's increase time
13. Biggest SCR's decrease time
14. Ratio of increase time and decrease time of the biggest SCR
15. Biggest SCR's duration (from peak start time until peak end time)
16. Biggest SCR's increase slope
17. Biggest SCR's decrease slope
18. Average amplitude increase of all SCRs
19. Average amplitude decrease of all SCRs
20. Average amplitude increase ratio of all SCRs
21. Average amplitude decrease ratio of all SCRs
22. Distance (duration) signal's max and min values
23. Overall signal change rate
24. Overall duration of the segments with a significant increase
25. Mean value of the segments with significant increase
26. Intensity of the segments with significant increase (duration x mean value)
27. Significant increase change (max – min of the segments with significant change)*
28. Significant increase divided by the duration (speed)
29. Overall duration of the segments with a significant decrease
30. Mean value of the segments with significant decrease
31. Intensity of the segments with significant decrease (duration x mean value)
32. Significant decrease change (max – min of the segments with significant change)*
33. Significant decrease divided by the duration (speed)

*Note – EDA significant increase/decrease: 10% change (increase or decrease) sustained for a duration of 15 seconds.

Table A.1: Features summary.

Blood Volume Pulse – 22 features	• 11 statistical features of the raw signal • 11 statistical features of the first derivative of the raw signal
Skin Temperature – 22 features	• 11 statistical features of the raw signal • 11 statistical features of the first derivative of the raw signal
Acceleration – 7 features	• 7 features: magnitude's mean value, top 3 frequencies; mean Power Spectral Density (PSD) of the top 3 frequencies
Heart Rate Variability Features – 20 features	• 10 HRV features based on BVP • 10 HRV features based on Empatica's IBI
Galvanic Skin Response – 61 features	• 11 statistical features of the raw signal • 11 statistical features of the first derivative of the raw signal • 39 SCR features

References

- [1] A. Alwan, "Global status report on noncommunicable diseases 2010," Geneva, Switzerland: World Health Organization; 2011. [Online]. Available: http://www.who.int/nmh/publications/ncd_report_full_en.pdf [Accessed 27.03.2017].
- [2] S. Karandeep and A. B. Landman, "Mobile health," in *Key Advances in Clinical Informatics*, pp. 183–196, Academic Press, 2017.
- [3] P. Cipresso, S. Serino, D. Villani, C. Repetto, L. Selitti, G. Albani, A. Mauro, A. Gaggioli, and G. Riva, "Is your phone so smart to affect your states? An exploratory study based on psychophysiological measures," *Neurocomputing*, vol. 84, pp. 23–30, 2012.
- [4] B. A. Borlaug, "Heart failure with preserved and reduced ejection fraction: Different risk profiles for different diseases," *European Heart Journal*, vol. 34, no. 19, pp. 34:19, 2013.
- [5] M. R. Cowie, "Clinical and economic burden of chronic heart failure," *Medicographia*, vol. 33, no. 3, pp. 370–376, 2011.
- [6] N. Azad and G. Lemay, "Management of chronic heart failure in the older population," *Journal of Geriatric Cardiology*, vol. 11, no. 4, p. 329, 2014.
- [7] A. E. Minton and L. R. Khale, "Belief systems, religion, and behavioral economics," New York: Business Expert Press LLC., 2014.
- [8] C. Marco and M. C. Varley, "Wearable training-monitoring technology: Applications, challenges, and opportunities," *International Journal of Sports Physiology and Performance*, vol. 2, no. 2, pp. S2–55, 2017.
- [9] D. Heaven, "Why faces don't always tell the truth about feelings," *Nature*, vol. 578, no. 7796, pp.502–504, 2020.
- [10] B. Felix, A. Paeschke, M. Rolfes, W. F. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *Proceedings of the European Conference on Speech Communication and Technology*, pp. 1517–1520, 2005.
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 1097–1105, 2012.
- [12] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *Proceedings of the Thirty-first AAAI Conference on Artificial intelligence*, pp. 4278–4284, 2017.

- [13] S. Chaib, H. Yao, Y. Gu, and M. Amrani, “Deep feature extraction and combination for remote sensing image classification based on pre-trained CNN models,” in *Proceedings of the International Conference on Digital Image Processing*, p. 104203D, 2017.
- [14] T. Vandal, E. Kodra, S. Ganguly, A. Michaelis, R. Nemani, and A. R., Ganguly, “DeepSD: Generating high resolution climate change projections through single image super-resolution,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1663–1672, 2017.
- [15] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.
- [16] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.
- [17] J. Devlin, M. W. Chang, M. K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018. arXiv preprint arXiv:1810.04805.
- [18] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, and S. Dieleman, “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484, 2016.
- [19] V. Oriol, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, and J. Oh, “Grandmaster level in StarCraft II using multi-agent reinforcement learning,” *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.
- [20] F. H. Ismail, G. Forestier, J. Weber, L. Idoumghar, and Pierre-Alain Muller, “Deep learning for time series classification: A review,” *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.
- [21] Fit4Work-Project: Self-management of physical and mental fitness of older workers. [Online]. Available: <http://www.fit4work-aal.eu/> [Accessed 21.07.2020].
- [22] STRAW-Project: Disentangling the sources and context of day-to-day stress at work. [Online]. Available: <https://strawproject.eu/> [Accessed 21.07.2020].
- [23] M. Gjoreski, M. Luštrek, M. Gams, and H. Gjoreski, “Monitoring stress with a wrist device using context,” *Journal of Biomedical Informatics*, vol. 73, pp. 159–170, 2017.
- [24] M. Gjoreski, H. Gjoreski, M. Luštrek, and M. Gams, “How accurately can your wrist device recognize daily activities and detect falls?,” *Sensors*, vol. 16, no. 6, p. 800, 2016.
- [25] M. Gjoreski, V. Janko, G. Slapničar, M. Mlakar, N. Reščič, J. Bizjak, V. Drobnič, M. Marinko, N. Mlakar, M. Luštrek, and M. Gams, “Classical, and deep learning methods for recognizing human activities, and modes of transportation with smartphone sensors,” *Information Fusion*, vol. 62, pp. 47–62, 2020.
- [26] V. Janko, N. Reščič, M. Mlakar, V. Drobnič, M. Gams, G. Slapničar, M. Gjoreski, J. Bizjak, M. Marinko, and M. Luštrek, “A new frontier for activity recognition: The

- Sussex-Huawei locomotion challenge,” in the UbiComp/ISWC'18 Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive, and Ubiquitous Computing, pp. 1511–1520, 2018.
- [27] M. Gjoreski, V. Janko, N. Reščič, M. Mlakar, M. Luštrek, J. Bizjak, G. Slapničar, M. Marinko, V. Drobnič, and M. Gams, “Applying multiple knowledge to Sussex-Huawei locomotion challenge,” in the UbiComp/ISWC'18 Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive, and Ubiquitous Computing, pp. 1488–1496, 2018.
- [28] V. Janko, M. Gjoreski, C. De Masi, N. Reščič, M. Luštrek, and M. Gams, “Cross-location transfer learning for the Sussex-Huawei locomotion recognition challenge,” in the UbiComp/ISWC'18 Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive, and Ubiquitous Computing, pp. 730–735, 2019.
- [29] H. Gjoreski, M. Ciliberto, L. Wang, F. J. O. Morales, S. Mekki, S. Valentin, and D. Roggen. “The University of Sussex-Huawei locomotion, and transportation dataset for multimodal analytics with mobile devices,” *IEEE Access*, vol. 6, pp. 42592–42604, 2018.
- [30] V. Janko, M. Gjoreski, G. Slapničar, M. Mlakar, N. Reščič, J. Bizjak, V. Drobnič, M. Marinko, N. Mlakar, M. Gams, and M. Luštrek, “Winning the Sussex-Huawei locomotion-transportation recognition challenge,” *Human Activity Sensing: Corpus, and Applications*, (Springer Series in Adaptive Environments), Cham: Springer, pp. 233–250, 2019.
- [31] H. Gjoreski, I. Kiprijanovska, S. Stankoski, S. Kalabakov, and M. Gjoreski, “Human activity recognition with head mounted IMU using weighted ensemble learning,” in *Proceedings of the International Conference on Activity and Behavior Computing*, accepted for publishing, 2020.
- [32] H. Gjoreski, S. Stankoski, I. Kiprijanovska, A. Nikolovska, N. Mladenovska, M. Trajanoska, B. Velichkovska, M. Gjoreski, M. Luštrek, and M. Gams, “Wearable sensors data-fusion, and machine-learning method for fall detection, and activity recognition,” *Challenges, and Trends in Multimodal Fall Detection for Healthcare*, (Studies in Systems, Decision, and Control, vol. 273), Cham: Springer, pp. 80–96, 2020.
- [33] M. Simjanoska, M. Gjoreski, M. Gams, and A. M. Bogdanova, “Non-invasive blood pressure estimation from ECG using machine learning techniques,” *Sensors*, vol. 18, no. 4, p. 1160, 2018.
- [34] M. Gjoreski, M. Simjanoska, A. Gradišek, A. M. Peterlin, M. Gams, and G. Poglajen, “Chronic heart failure detection from heart sounds using a stack of machine-learning classifiers,” in *Proceedings of the International Conference on Intelligent Environments*, pp. 14–19, 2017.
- [35] T. Knez, “Analyzing effect of computer game difficulty on biological signals,” Bachelor’s Thesis, First Cycle University Study Programme Computer and Information Science, University of Ljubljana, Ljubljana, 2019.

- [36] T. Knez, M. Gjoreski, and V. Pejović, “Analiza vpliva težavnosti računalniške igre na izmerjene vrednosti fizioloških signalov,” in Proceedings of the International Multiconference Information Society, pp. 5–8, 2019.
- [37] M. Gjoreski, T. Kolenik, T. Knez, M. Luštrek, M. Gams, H. Gjoreski, and V. Pejović, “Datasets for cognitive load inference using wearable sensors, and psychological traits,” *Applied Sciences*, vol. 10, no. 11, p. 3843, 2020.
- [38] M. Gjoreski, M. Luštrek, and M. Gams, “Multi-task ensemble learning for affect recognition,” in the UbiComp/ISWC'19 Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive, and Ubiquitous Computing, pp. 553–558, 2018.
- [39] M. Gjoreski, S. Kalabakov, M. Luštrek, M. Gams, and H. Gjoreski, “Cross-dataset deep transfer learning for activity recognition,” in the UbiComp/ISWC'19 Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 714–718, 2019.
- [40] M. Gjoreski, B. Mitrevski, M. Luštrek, and M. Gams, “An inter-domain study for arousal recognition from physiological signals,” *Informatica*, vol. 42, no. 1, pp. 61–68, 2018.
- [41] M. Gjoreski, A. Gradišek, B. Budna, M. Gams, and G. Poglajen, “Machine learning and end-to-end deep learning for the detection of chronic heart failure from heart sounds,” *IEEE Access*, vol. 8, pp. 20313–20324, 2020.
- [42] G. Slapničar, N. Mlakar, and M. Luštrek, “Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network,” *Sensors*, vol. 19, no. 15, p. 3420, 2019.
- [43] M. Gjoreski, M. Gams, M. Luštrek, P. Genc, J.-U. Garbas, and T. Hassan, “Machine learning and end-to-end deep learning for monitoring driver distractions from physiological and visual signals,” *IEEE Access*, vol. 8, pp. 70590–70603, 2020.
- [44] C. D. Norman and H. A. Skinner, “eHEALS: The eHealth literacy scale,” *Journal of Medical Internet Research*, vol. 8 no. 4, p. e27, 2006.
- [45] D. Tara, K. Petrie, J. Proudfoot, J. Clarke, M.-R. Birch, and H. Christensen, “Smartphones for smarter delivery of mental health programs: A systematic review,” *Journal of Medical Internet Research*, vol. 15, no. 11, p. e247, 2013.
- [46] D. D. Luxton, R. A. McCann, N. E. Bush, M. C. Mishkind, and G. M. Reger, “mHealth for mental health: Integrating smartphone technology in behavioral healthcare,” *Professional Psychology: Research and Practice*, vol. 42, no. 6, p. 505, 2011.
- [47] C. L. Ventola, “Mobile devices and apps for health care professionals: uses and benefits,” *Pharmacy and Therapeutics*, vol. 39, no. 5, p. 356, 2014.
- [48] B. Martínez-Pérez, I. De La Torre-Díez, and M. López-Coronado, “Mobile health applications for the most prevalent conditions by the World Health Organization: Review and analysis,” *Journal of Medical Internet Research*, vol. 15, no. 6, p. 120, 2013.

- [49] H. Hongwei, X. Youzhi, Y. Hairong, S. Mubeen, and Z. Hongke, "An elderly health care system using wireless sensor networks at home," in *Proceedings of the Conference on Sensor Technologies and Applications*, pp. 158–163, 2009.
- [50] J. Ishio and N. Abe, "Measuring affective well-being by the combination of the day reconstruction method and a wearable device: Case study of an aging and depopulating community in Japan," *Augmented Human Research*, vol. 2, no. 1, p. 2, 2017.
- [51] M. Rabbi, S. Ali, T. Choudhury, and E. Berke, "Passive and in-situ assessment of mental and physical well-being using mobile sensors," in *Proceedings of the ACM Conference on Ubiquitous Computing*, pp. 385–394, 2011.
- [52] R. Wang et al., "StudentLife: Assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proceedings of the ACM Conference on Ubiquitous Computing*, pp. 3–14, 2014.
- [53] G. M. Harari, S. D. Gosling, R. Wang, F. Chen, Z. Chen, Z., and A. T. Campbell, "Patterns of behavior change in students over an academic term: A preliminary study of activity and sociability behaviors using smartphone sensing methods," *Computers in Human Behavior*, vol. 67, pp. 129–138, 2017.
- [54] R. Shriram, A. Wakankar, N. Daimiwal, and D. Ramdasi, "Continuous cuffless blood pressure monitoring based on PTT," in *Proceedings of the Bioinformatics and Biomedical Technology Conference, IEEE*, pp. 51–55, 2010.
- [55] A. Sahoo, P. Manimegalai, and K. Thanushkodi, "Wavelet based pulse rate and blood pressure estimation system from ECG and PPG signals," in *Proceedings of the Computer, Communication and Electrical Technology Conference, IEEE*, pp. 285–289, 2011.
- [56] S.S. Thomas, V. Nathan, C. Zong, K. Soundarapandian, X. Shi, and R. Jafari, "BioWatch: A noninvasive wrist-based blood pressure monitor that incorporates training techniques for posture and subject variability," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, pp. 1291–1300, 2016.
- [57] R. Nye, Z. Zhang, and Q. Fang, "Continuous non-invasive blood pressure monitoring using photoplethysmography: A review," in *Proceedings of the Bioelectronics and Bioinformatics Symposium*, pp. 176–179, 2015.
- [58] S. Goli and T. Jayanthi, "Cuff less continuous non-invasive blood pressure measurement using pulse transit time measurement," *International Journal of Recent Development in Engineering and Technology*, vol. 2, no. 87, pp. 16–86, 2014.
- [59] J. Seo, S.J. Pietrangelo, H.S. Lee, and C.G. Sodini, "Noninvasive arterial blood pressure waveform monitoring using two-element ultrasound system," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 62, pp. 776–784, 2015.
- [60] M.Y.M. Wong, C.C.Y. Poon, and Y.T. Zhang, "An evaluation of the cuffless blood pressure estimation based on pulse transit time technique: A half year study on normotensive subjects," *Cardiovascular Engineering*, vol. 9, no. 1, pp. 32–38, 2009.
- [61] R. Payne, C. Symeonides, D. Webb, and S. Maxwell, "Pulse transit time measured from the ECG: An unreliable marker of beat-to-beat blood pressure," *Journal of Applied Physiology*, vol. 100, no. 1, pp. 136–141, 2006.

- [62] K. Chan, K. Hung, and Y. Zhang, “Noninvasive and cuffless measurements of blood pressure for telemedicine,” *Engineering in Medicine and Biology Society*, in *Proceedings of the 23rd Annual International Conference of the IEEE*, vol. 4, pp. 3592–3593, 2001.
- [63] S. Ahmad, S. Chen, K. Soueidan, I. Batkin, M. Bolic, H. Dajani, and V. Groza, “Electrocardiogram-assisted blood pressure estimation,” *IEEE Transactions on Biomedical Engineering*, vol. 59, pp. 608–618, 2012.
- [64] E.B. Schroeder, D. Liao, L.E. Chambless, R.J. Prineas, G.W. Evans, and G. Heiss, “Hypertension, blood pressure, and heart rate variability,” *Hypertension*, vol. 42, no. 6, pp. 1106–1111, 2003.
- [65] M.K.B.A. Hassan, M. Mashor, N.M. Nasir, and S. Mohamed, “Measuring of systolic blood pressure based on heart rate,” in *Proceedings of the 4th Kuala Lumpur Conference on Biomedical Engineering*, Springer, pp. 595–598, 2008.
- [66] S.C. Segerstrom and G.E. Miller, “Psychological stress and the human immune system: a meta-analytic study of 30 years of inquiry,” *Psychological Bulletin*, vol. 130, no. 4, p. 601, 2004.
- [67] H. G. Ice and G. D. James, “Measuring stress in humans: A practical guide for the field,” Cambridge University Press, 2007.
- [68] E. Oetter, “Heart rate variability research review,” [Online]. Available: <http://www.8weeksoout.com/2011/12/05/heart-rate-variability-research-review/>, [Accessed 27.03.2017].
- [69] W.B. Cannon, “The wisdom of the body,” W.W. Norton, New Work, 1932.
- [70] J. A. Healey and R. W. Picard, “Detecting stress during real-world driving tasks using physiological sensors,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 156–166, 2005.
- [71] M. van Dooren, J.J.G. de Vries, and J. H. Janssen, “Emotional sweating across the body: Comparing 16 different skin conductance measurement locations,” *Physiology Behavior*, vol. 106, no. 2, pp. 298–304, 2012.
- [72] H. Sarker et al., “Finding significant stress episodes in a discontinuous time series of rapidly varying mobile sensor data,” in *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 4489–4501, 2016.
- [73] K. Hovsepian, M. Absi, T. Kamarek, and M. Nakajima, “cStress: Towards a gold standard for continuous stress assessment in the mobile environment,” in *Proceedings of the ACM Conference on Ubiquitous Computing*, pp. 493–504, 2015.
- [74] F.G. Paas, and J.J. Van Merriënboer, “Instructional control of cognitive load in the training of complex cognitive tasks,” *Educational Psychology Review*, vol. 6, pp. 351–371, 1994.
- [75] T. Matkovič and V. Pejović “Wi-mind: Wireless mental effort inference,” in *Proceedings of the ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, Singapore, pp. 1241–1249, 2018.
- [76] C. Setz, B. Arnrich, J. Schumm, R. La Marca, G. Tröster, and U. Ehlert, “Discriminating stress from cognitive load using a wearable EDA device,” *IEEE Transactions on Information Technologies in Biomedicine*, vol. 14, pp. 410–417, 2009.

- [77] G. J. K. Novak, K. Stojmenova, and J. Sodnik, "Assessment of cognitive load through biometric monitoring," Society for Information Systems and Computer Networks: Ljubljana, Slovenia, 2017.
- [78] F. Schaule, J.O. Johanssen, B. Bruegge, and V. Loftness, "Employing consumer wearables to detect office workers' cognitive load for interruption management," in Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 2, no. 1, pp. 1–32, 2018.
- [79] E. Haapalainen, S. Kim, J.F. Forlizzi, and A.K. Dey, "Psycho-physiological measures for assessing cognitive load," in Proceedings of the ACM International Conference on Ubiquitous Computing, pp. 301–310, 2010.
- [80] M. El Ayadi, M. S. Kamel, and F. Karay, "Survey on speech emotion recognition: Features, classification schemes and databases," Pattern Recognition, vol. 44, no. 3 pp. 572–587, 2011.
- [81] Microsoft's API for emotion recognition from images, [Online], Available: <https://www.microsoft.com/cognitive-services/en-us/emotion-api>, [Accessed 28.03.2017].
- [82] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE - The munich versatile and fast open-source audiomfeature extractor," in Proceedings of the ACM International Conference on Multimedia, pp. 1459–1462, 2010.
- [83] G. Trigeorgis et al., "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," in Proceedings of the Conference on Acoustics, Speech and Signal Processing, pp. 5200–5204, 2016.
- [84] R. Subramanian, J. Wache, M. Abadi, R. Vieriu, S. Winkler, and N. Sebe, "ASCERTAIN: Emotion and personality recognition using commercial sensors," IEEE Transactions on Affective Computing, vol. 9, no. 2, pp. 147–160 2016.
- [85] S. Koelstra, C. Muehl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "DEAP: A database for emotion analysis using physiological signals," IEEE Transaction on Affective Computing, vol. 3, no. 1, pp. 18–31, 2012.
- [86] D. Iacovielloa, A. Petraccab, M. Spezialettib, and G. Placidib, "A real-time classification algorithm for EEG-based BCI driven by self-induced emotions," Computer Methods and Programs in Biomedicine, vol. 122, no. 3, pp. 293–303, 2015.
- [87] M. Khezria, M.Firoozabadib, and A. R. Sharafata, "Reliable emotion recognition system based on dynamic adaptive fusion of forehead biopotentials and physiological signals," Computer Methods and Programs in Biomedicine, vol. 122, no. 2, pp. 149–164, 2015.
- [88] R. M. Mehmooda and H. J. Leea, "A novel feature extraction method based on late positive potential for emotion recognition in human brain signal patterns," Computers and Electrical Engineering, vol. 53, pp. 444–457, 2016.
- [89] A. Mikuckas, I. Mikuckiene, A. Venckauskas, E. Kazanavicius, R. Lukas, and I. Plauska, "Emotion recognition in human computer interaction systems," Elektronika ir Elektrotechnika, Reserach Journal, vol. 20, no. 10, pp. 51–56, 2014.
- [90] Z. Yin, M. Zhao, Y. Wang, J. Yang, and J. Zhang, "Recognition of emotions using multimodal physiological signals and an ensemble deep learning model," Computer Methods and Programs in Biomedicine, vol. 140, pp. 93–110, 2017.

- [91] J. M. Kelly, R. E. Strecker, and M. T. Bianchi, "Recent developments in home sleep-monitoring devices," International Scholarly Research Network, ISRN Neurology, 2012.
- [92] P. Ekman, W.V. Friesen, and P. Ellsworth, "Emotion in the human face," Studies in Emotion and Social Interaction, Pergamon Press, 2nd Edition, 1972.
- [93] J. A. Russell, "A circumplex model of affect," Journal of Personality and Social Psychology, vol. 39, no. 6, p.1161 1980.
- [94] D. G. Myers, "Theories of emotion," in Psychology: Seventh Edition, New York, NY: Worth Publishers 500, 2004.
- [95] E. Garcia-Ceja, M. Riegler, T. Nordgreen, P. Jakobsen, K. J. Oedegaard, and J. E. Tørresen, "Mental health monitoring with multimodal sensing and machine learning: A survey," Pervasive and Mobile Computing, vol. 51, pp. 1–26, 2018.
- [96] A. Pittig, J. J. Arch, C. W. Lam, and M. G. Craske, "Heart rate and heart rate variability in panic, social anxiety, obsessive-compulsive, and generalized anxiety disorders at baseline and in response to relaxation and hyperventilation," International Journal of Psychophysiology, vol. 87, no. 1, pp.19–27, 2013.
- [97] M. Cella, L. Okruszek, M. Lawrence, V. Zarlenga, Z. He, and T. Wykes, "Using wearable technology to detect the autonomic signature of illness severity in schizophrenia," Schizophrenia Research, vol. 195, pp. 537–542, 2018.
- [98] Y. Tazawa, L. Kuo-ching, Y. Michitaka, K. Momoko, K. Yuriko, T. Akihiro, K. Aiko et al., "Evaluating depression with multimodal wristband-type wearable device: Screening and assessing patient severity utilizing machine-learning," Heliyon, vol. 6, no. 2, pp. 1–26, 2020.
- [99] B. K. Pedersen and B. Saltin, "Evidence for prescribing exercise as therapy in chronic disease," Scandinavian Journal of Medicine and Science in Sports, vol. 16, no. 1, pp. 3–63, 2006.
- [100] G. Plasqui and K. R. Westerterp, "Physical activity assessment with accelerometers: An evaluation against doubly labeled water," Obesity, vol. 15, pp. 2371–2379, 2007.
- [101] H. Gjoreski, B. Kaluža, M. Gams, R. Milić, and M. Luštrek, "Ensembles of multiple sensors for human energy expenditure estimation," in Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 359–362, 2013.
- [102] S. Masood, A. Rai, A. Aggarwal, M. N. Doja, and M. Ahmad, "Detecting distraction of drivers using convolutional neural network," Pattern Recognition Letters, vol. 139, pp. 79–85, 2018.
- [103] F. Eyben et al., "Emotion on the road—necessity, acceptance, and feasibility of affective computing in the car," Advances Human-Computer Interactions, vol. 2010, pp. 1–17, 2010.
- [104] M. A. Regan, C. Hallett, and C. P. Gordon, "Driver distraction and driver inattention: Definition, relationship and taxonomy," Accident Analysis and Prevention, vol. 43, no. 5, pp. 1771–1781, 2011.
- [105] T. W. Strine and D. P. Chapman, "Associations of frequent sleep insufficiency with health-related quality of life and health behaviors," Sleep Medicine, vol. 6, no. 1, pp. 23–27, 2005.

- [106] A. Sathyanarayana et al., "Sleep quality prediction from wearable data using deep learning," *JMIR mHealth and uHealth*, vol. 4, p. e125, 2016.
- [107] D. B. Springer, L. Tarassenko, and G. D. Clifford, "Logistic regression-HSSM-based heart sound segmentation," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 4, pp. 822–832, 2015.
- [108] M. N. Homsy et al., "Automatic heart sound recording classification using a nested set of ensemble algorithms," in *Proceedings of Computing in Cardiology Conference*, pp. 817–820, 2016.
- [109] F. Plesinger et al., "Heart sounds analysis using probability assessment," *Physiological Measurement*, vol. 38, no. 8, p. 1685, 2017.
- [110] Sh. Dvijesh, M. Papadakis, P. Tsiamyrtzis, B. Bass, and I. Pavlidis, "Perinatal imaging of physiological stress and its affective potential," *IEEE Transactions on Affective Computing* vol. 3, no. 3, pp. 366–378, 2012.
- [111] D. McDuff, S. Gontarek, and R. Picard, "Remote measurement of cognitive stress via heart rate variability," in *Proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2957–2960, 2014.
- [112] K. M. van der Kooij and M. Naber "An open-source remote heart rate imaging method with practical apparatus and algorithms," *Behavior Research Methods*, vol. 51, no. 5, pp. 2106–2119, 2019.
- [113] M. N. Mohammed, H. Syamsudin, S. Al-Zubaidi, R. R. Aks, and E. Yusuf, "Novel COVID-19 detection and diagnosis system using IoT based smart helmet," *International Journal of Psychosocial Rehabilitation*, vol. 24, no. 7, pp. 2296–2303, 2020.
- [114] E. Alpaydin, "Introduction to machine learning," MIT press, 2020.
- [115] EU report on "Amenable and preventable deaths statistics," [Online]. Available: http://ec.europa.eu/eurostat/statistics-explained/index.php/Amenable_and_preventable_deaths_statistics, [Accessed 30.03.2017].
- [116] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, pp. 144–156, 2018.
- [117] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.
- [118] M. Hazewinkel, "Minimax principle," in *Encyclopedia of Mathematics*, Springer Science, Business Media, B.V., Kluwer Academic Publishers, 1994.
- [119] G. Wiederhold and J. McCarthy; Ed Feigenbaum, "Memorial resolution: Arthur L. Samuel," *Stanford University Historical Society*, 1990.
- [120] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [121] N. J. Nilsson, "Learning Machines," McGraw-Hill Companies, First Edition, 1965.

- [122] G. S. Sebestyen, "Decision-making processes in pattern recognition," Macmillan, First Printing Edition, 1962.
- [123] J. R. Quinlan, "Discovering rules by induction from large collections of examples," in *Expert Systems in the Microelectronics Age*, Edinburgh University Press, 1979.
- [124] J. R. Quinlan, "C4.5: Programs for machine learning," Morgan Kaufmann Publishers, 1993.
- [125] R. E. Schapire, "The strength of weak learnability," *Machine Learning*, vol. 5, no. 2, pp. 197–227, 1990.
- [126] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [127] C. Cortes and V. N. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20 no. 3, pp. 273–297, 1995.
- [128] T. K. Ho, "Random decision forests," in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 278–282, 1995.
- [129] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [130] Y. LeCun, "The deep learning – applied math connection", presentation at the Society for Industrial and Applied Mathematic, US 2020. [Online]. Available: https://bytes.usc.edu/cs585/m20_dBdSmLdM/extras/docs/LeCun_MDS20.pdf [Accessed 28.10.2020].
- [131] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [132] H. J. Kelley, "Gradient theory of optimal flight paths," *ARS Journal*, vol. 30, no. 10, pp. 947–954, 1960.
- [133] A. E. Bryson, "A gradient method for optimizing multi-stage allocation processes," in *Proceedings of the Harvard University Symposium on Digital Computers and Their Applications*, 1961.
- [134] A. Bryson and Y. Ho, "Applied optimal control: optimization, estimation, and control," Blaisdell Pub. Co, 1969
- [135] S. Linnainmaa, "The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors," Master's Thesis, Univ. Helsinki, 1970.
- [136] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing*, vol. 1, pp. 318–362. MIT Press, 1986.
- [137] K. Fukushima, "Neural network model for a mechanism of pattern recognition unaffected by shift in position – Neocognitron," *IEICE Technical Report*, vol. 62, no. 10, pp. 658–665, 1979.
- [138] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R.E Howard, W. Hubbard, and L. D. Jackel, "Back-propagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp.541–551, 1989.

- [139] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [140] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.
- [141] J. Z. Sasiadek and P. Hartana, “Sensor data fusion using Kalman filter,” in *Proceedings of the Third International Conference on Information Fusion*, pp. 19–25, 2000.
- [142] C. Giraud and B. Jouvencel, “Sensor Selection: A Geometrical Approach,” in *Proceedings of the International Conference on Intelligent Robots and Systems*, vol. 2, pp. 555–560, 1995.
- [143] K. Marzullo, “Tolerating Failures of Continuous-Valued Sensors,” in *Proceedings of the ACM Transactions on Computer Systems*, vol. 8, no. 4, pp. 284–304, 1990.
- [144] F. Russo and G. Ramponi, “Fuzzy methods for multisensor data fusion,” *IEEE Transactions on Instrumentation and Measurement*, vol. 43, no. 2, pp. 288–294, 1994.
- [145] H. Hongwei, X. Youzhi, Y. Hairong, S. Mubeen, and Z. Hongke, “An elderly health care system using wireless sensor networks at home,” *Sensor Technologies and Applications*, vol. 1, pp. 158–163, 2009.
- [146] H. Gjoreski, “Context-based reasoning in ambient intelligence,” PhD Thesis, Jožef Stefan International Postgraduate School, Ljubljana, Slovenia, 2015.
- [147] J. A. Ward, D. Richardson, G. Orgs, K. Hunter, and A. Hamilton, “Sensing interpersonal synchrony between actors and autistic children in theatre using wrist-worn accelerometers,” in *Proceedings of the ACM International Symposium on Wearable Computers*, pp. 148–155, 2018.
- [148] S. Choi and Z. Jiang, “Comparison of envelope extraction algorithms for cardiac sound signal segmentation,” *Expert Systems with Applications*, vol. 34, no. 2, pp. 1056–1069, 2008.
- [149] S. Butterworth, “On the theory of filter amplifiers,” *Wireless Engineer*, vol. 7, no. 6, pp. 536–541, 1930.
- [150] N.R. Lomb, “Least-squares frequency analysis of unequally spaced data,” *Astrophysics and Space Science*, vol. 39, pp. 447–462, 1976.
- [151] R. Castaldo, P. Melillo, U. Bracale, M. Caserta, M. Triassi, and L. Pecchia, “Acute mental stress assessment via short term HRV analysis in healthy adults: A systematic review with meta-analysis,” *Biomedical Signal Processing and Control*, vol. 18, pp. 370–377, 2015.
- [152] R.A. Hoshi, C.M. Pastre, L.C.M. Vanderlei, and M.F. Godoy, “Poincaré plot indexes of heart rate variability relationships with other nonlinear variables,” *Autonomic Neuroscience*, vol. 177, pp. 271–274, 2013.
- [153] M. M. Bradley and P. J. Lang, “Affective reactions to acoustic stimuli,” *Psychophysiology*, vol. 37, no. 2, pp. 204–215, 2000.

- [154] H. Critchley, “Electrodermal responses: What happens in the brain,” *The Neuroscientist*, vol. 8, no. 2, pp. 132–142, 2002.
- [155] R. W. Picard, S. Fedor, and Y. Ayzenberg, “Multiple arousal theory and daily-life electrodermal activity asymmetry,” *Emotion Review*, vol. 8, no. 1, pp. 62–75, 2016.
- [156] W. Boucsein, “Electrodermal activity,” Springer Science & Business Media, 2012.
- [157] S. Taylor, N. Jaques, W. Chen, S. Fedor, A. Sano, and R. Picard, “Automatic identification of artifacts in electrodermal activity data,” in *Proceedings of the Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1934–1937, 2015.
- [158] R. Tibshirani, “Regression shrinkage and selection via the LASSO,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–88, 1996.
- [159] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *Proceedings of the International Conference on Machine Learning*, pp. 609–616, 2009.
- [160] F. Ordonez and D. Roggen, “Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, no. 1, p. 115, 2016.
- [161] C. Szegedy, S. Ioffe, and V. Vanhoucke, “Inception-v4, Inception-ResNet and the impact of residual connections on learning,” in *Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence*, pp. 4278–4284 2017.
- [162] M. Gams, “Weak intelligence: Through the principle and paradox of multiple knowledge,” Nova Science, 2001.
- [163] F. E. Jr. White, “Data fusion lexicon,” Joint Directors of Laboratories, Technical Panel for C3, Data Fusion Sub-Panel, Naval Ocean Systems Center, San Diego, 1987.
- [164] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, pp. 41–75, 1997.
- [165] J. West, D. Ventura, and S. Warnick, “Spring research presentation: A theoretical foundation for inductive transfer,” Brigham Young University, College of Physical and Mathematical Sciences, 2007.
- [166] S. Pan and Q. Yang, “A survey on transfer learning,” *Knowledge Data Entering IEEE Transactions*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [167] F. Shaffer, R. McCraty, and C. L. Zerr, “A healthy heart is not a metronome: An integrative review of the heart’s anatomy and heart rate variability,” *Frontiers in Psychology*, vol. 5, p. 1040, 2014.
- [168] R. Wang and K. Tang, “Feature selection for maximizing the area under the ROC curve,” in *Proceedings of the IEEE International Conference on Data Mining Workshops*, pp. 400–405, 2009.
- [169] G. Xuan et al., “Feature selection based on the Bhattacharyya distance,” in *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 957–957, 2006.
- [170] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” *Physical Review E*, vol. 69, no. 6, p. 066138, 2004.

- [171] C. Arndt, “Information measures: Information and its description in science and engineering,” Springer Science and Business Media, 2012.
- [172] Q. Gu, Z. Li, and J. Han, “Generalized Fisher score for feature selection,” arXiv preprint arXiv:1202.3725, 2012.
- [173] R. F. Woolson, “Wilcoxon signed-rank test,” Wiley Encyclopedia of Clinical Trials, pp. 1–3, 2007.
- [174] J.R. Quinlan, “Improved use of continuous attributes in c4.5,” *Journal on Artificial Intelligence Research*, vol. 4, pp. 77–90, 1996.
- [175] S. Russel and P. Norvig, “Artificial Intelligence: A modern approach,” 2nd Ed., Prentice Hall: New York, USA, 2003.
- [176] D. Aha and D. Kibler, “Instance-based learning algorithms,” *Machine Learning*, vol. 6, pp. 37–66, 1991.
- [177] N. Cristianini and J. Shawe-Taylor, “An introduction to support vector machines and other kernel-based learning methods,” Cambridge University Press: Cambridge, UK, 2000.
- [178] L. Breiman, “Bagging predictors,” Technical report No. 421, University of California, 1994.
- [179] K. Michael, “Thoughts on hypothesis boosting,” Unpublished manuscript, December 1988.
- [180] H. Tin Kam, “Random decision forests,” in *Proceedings of the International Conference on Document Analysis and Recognition*, pp. 278–282, 1995.
- [181] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” *Advances in Neural Information Processing Systems*, pp. 2546–2554, 2011.
- [182] C. Setz, J. Schumm, C. Lorenz, B. Arnrich, and F. Tröster, “Combining worthless sensor data,” in *Proceedings of the Measuring Mobile Emotions Workshop*, 2009.
- [183] K. Petr, B. Gabrys, and S. Strandt. “Data-driven soft sensors in the process industry,” *Computers and Chemical Engineering*, vol. 33, no. 4, pp. 795–814, 2009.
- [184] K. Sun, J. Liu, J. L. Kang, S. S. Jang, D. S. H Wong, and D. S. Chen, “Soft sensor development with nonlinear variable selection using nonnegative garrote and artificial neural network,” *Computer Aided Chemical Engineering*, vol. 33, pp. 883–888, 2014.
- [185] D. Ravi, C. Wong, B. Lo, and G. Z. Yang, “Deep learning for human activity recognition: A resource efficient implementation on low-power devices,” in *Proceedings of the International Conference on Wearable and Implantable Body Sensor Networks*, pp. 71–76, 2016.
- [186] K. Hovsepian, M. Al'Absi, E. Ertin, T. Kamarck, M. Nakajima, and S. Kumar, “cStress: Towards a gold standard for continuous stress assessment in the mobile environment,” in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 493–504, 2015.

- [187] S. Koelstra, M. Soleymani, A. Yazdani, T. Ebrahimi, A. Nijholt, and I. Patras, "DEAP: A database for emotion analysis using physiological signals," *IEEE Transactions on Affective Computing*, vol. 3, no. 1, pp. 18–31, 2012.
- [188] M. Gjoreski, H. Gjoreski, and A. Kulakov, "Machine learning approach for emotion recognition in speech," *Informatica*, vol. 38, no. 4, pp. 377–384, 2014.
- [189] R. Subramanian, J. Wache, M. Abadi, R. Vieriu, S. Winkler, and N. Sebe, "ASCERTAIN: Emotion and Personality Recognition using Commercial Sensors," *IEEE Transactions on Affective Computing*, vol. 9, no. 2, pp. 147–160, 2016.
- [190] S. Koelstra, C. Muehl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "DEAP: A database for emotion analysis using physiological signals," *IEEE Transaction on Affective Computing*, vol. 3, no. 1, pp. 18–31, 2012.
- [191] S. Schneegass, B. Pfleging, N. Broy, A. Schmidt, and F. Heinrich, "A data set of real world driving to assess driver workload," in *Proceedings of the International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pp. 150–157, 2013.
- [192] M. Soleymani and T. Pun, "A multimodal database for affect recognition and implicit tagging," *IEEE Transactions on Affective Computing*, vol. 3, no. 1, pp. 42–55, 2012.
- [193] J. A. Miranda-Correa, M. Khomami Abadi, N. Sebe, and I. Patras, "Amigos: a dataset for affect, personality and mood research on individuals and groups," *IEEE Transactions on Affective Computing*, 2017.
- [194] T. Knez, "Analyzing effect of computer game difficulty on biological signals," Bachelor's Thesis, First Cycle University Study Programme Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia, 2019.
- [195] V. Janko, "Adapting sensor settings for energy-efficient context recognition," PhD Thesis, Jozef Stefan International Postgraduate School, Ljubljana, Slovenia, 2020.
- [196] S. Karandeev and A. B. Landman, "Mobile health," in *Key Advances in Clinical Informatics*, pp. 183–196. Academic Press, 2017.
- [197] P. Cipresso, S. Serino, D. Villani, C. Repetto, L. Selitti, G. Albani, A. Mauro, A. Gaggioli, and G. Riva, "Is your phone so smart to affect your states? An exploratory study based on psychophysiological measures," *Neurocomputing*, vol. 84, pp. 23–30, 2012.
- [198] M. A. Hanson, et al., "Body area sensor networks: Challenges and opportunities," *Computer*, vol. 42, pp. 158–65, 2009.
- [199] T. Hsiao-Yu, H.-W. Tung, Er. Yumer, K. Fragkiadaki. "Self-supervised learning of motion capture," *Advances in Neural Information Processing Systems*, pp. 5236–5246. 2017.
- [200] S. Wachter, M. Brent, and C. Russel, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard Journal of Law and Technology*, vol. 31, p. 841, 2017.

- [201] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.
- [202] S. Joshi, K. Oluwasanmi, V. Warut, K. Been, and G. Joydeep, “Towards realistic individual recourse and actionable explanations in black-box decision making systems,” *arXiv preprint arXiv:1907.09615*, 2019.
- [203] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, and R. Chatila, “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Information Fusion*, vol. 58, pp. 82–115, 2020.
- [204] K. Crawford and V. Joler, “Anatomy of an AI system,” [Online], Available: <https://anatomyof.ai/> [Accessed 29.10.2020]

Bibliography

Publications Related to the Thesis

Journal Articles

- M. Gjoreski, V. Janko, G. Slapničar, M. Mlakar, N. Reščič, J. Bizjak, V. Drobnič, M. Marinko, N. Mlakar, M. Luštrek, and M. Gams, "Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors," *Information Fusion*, vol. 62, pp. 47–62, 2020. **IF: 13.669, (A", 1/4).**
- M. Gjoreski, A. Gradišek, B. Budna, M. Gams, and G. Poglajen, "Machine learning and end-to-end deep learning for the detection of chronic heart failure from heart sounds," *IEEE Access*, vol. 8, pp. 20313–20324, 2020. **IF: 4.64, (A', 1/4).**
- M. Gjoreski, M. Gams, M. Luštrek, P. Genc, J.-U. Garbas, and T. Hassan, "Machine learning and end-to-end deep learning for monitoring driver distractions from physiological and visual signals," *IEEE Access*, vol. 8, pp. 70590–70603, 2020. **IF: 4.64, (A', 1/4).**
- V. Pejović, M. Gjoreski, C. Anderson, K. David, and M. Luštrek, "Toward cognitive load inference for attention management in ubiquitous systems," *IEEE Pervasive Computing: Mobile and Ubiquitous Systems*, vol. 19, no. 2, pp. 35–45, 2020. **IF: 3.02, (A', 1/4).**
- M. Gjoreski, T. Kolenik, T. Knez, M. Luštrek, M. Gams, H. Gjoreski, and V. Pejović, "Datasets for cognitive load inference using wearable sensors and psychological traits," *Applied Sciences*, vol. 10, no. 1, 3843, 2020. **IF: 2.474, (A^{1/2}, 2/4).**
- M. Simjanoska, M. Gjoreski, M. Gams, and A. M. Bogdanova, "Non-invasive blood pressure estimation from ECG using machine learning techniques," *Sensors*, vol. 18, no. 4, pp. 1160–20, 2018. **IF: 3.275, (A', 1/4).**
- M. Gjoreski, B. Mitrevski, M. Luštrek, and M. Gams, "An inter-domain study for arousal recognition from physiological signals," *Informatika*, vol. 42, no. 1, pp. 61–68, 2018.
- M. Gjoreski, M. Luštrek, M. Gams, H. Gjoreski, "Monitoring stress with a wrist device using context," *Journal of Biomedical Informatics*, vol. 73, pp. 159–170, 2017. **IF: 3.526, (A^{1/2}, 2/4).**
- M. Gjoreski, H. Gjoreski, M. Luštrek, and M. Gams, "How accurately can your wrist device recognize daily activities and detect falls?," *Sensors*, vol. 16, no. 6, 2016. **IF: 3.275, (A', 1/4).**

Conference Papers

- S. Gashi, E. Di Lascio, B. stancu, V. Das Swain, M. Gjoreski, V. Mishra, and S. Santini, "Automatic detection of artifacts in electrodermal activity sensor data," in *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, pp. 1–31, 2020.

- H. Gjoreski, I. Kiprijanovska, S. Stankoski, S. Kalabakov, and M. Gjoreski, "Human activity recognition with head mounted imu using weighted ensemble learning," in Proceedings of the Conference on Activity and Behavior Computing, accepted for publishing, 2020.
- C. Picard, V. Janko, N. Reščič, M. Gjoreski, and M. Luštrek, "Identification of cooking preparation using motion capture data: A submission to the cooking activity recognition challenge," in Proceedings of the Conference on Activity and Behavior Computing, accepted for publishing, 2020.
- T. Knez, M. Gjoreski, and V. Pejović, "Analiza vpliva težavnosti računalniške igre na izmerjenevrednosti fizioloških signalov," in Proceedings of the International Multiconference Information Society, pp. 5–9, 2019.
- M. Gjoreski, A. Gradišek, B. Budna, and G. Poglajen, "Feature analysis for detecting different chronic heart failure stages," in Proceedings of the International Multiconference Information Society, pp. 9–12, 2019.
- T. Kolenik, M. Gjoreski, and M. Gams "Designing an intelligent cognitive assistant for behavior change in mental health," in Proceedings of the International Multiconference Information Society, pp. 69–72, 2019.
- J. Lukan, M. Gjoreski, H. Mauersberger, A. Hoppe, U. Hess, and M. Luštrek, "Analysing physiology of interpersonal conflicts using a wrist device," in Proceedings of the 14th European Conference on Ambient Intelligence, pp. 162–167, 2018.
- B. Mitrevski, V. Petreski, M. Gjoreski, and B. R. Stojkosks, "Framework for human activity recognition on smartphones and smartwatches," in Proceedings of the International Conference, ICT Innovations, pp. 90–99, 2018.
- M. Simjanoska, M. Gjoreski, M. Gams, Matjaž, and A. M. Bogdanova, "Novel data processing approach for deriving blood pressure from ECG only," in Proceedings of the 10th International Conference, ICT Innovations, pp. 273–285, 2018.
- M. Trajanoska, M. Katrašnik, J. Lukan, M. Gjoreski, H. Gjoreski, and M. Luštrek, "Context-Aware Stress Detection In The Aware Framework," in Proceedings of the International Multiconference Information Society, pp. 25–28, 2018.
- M. Gjoreski, M. Simjanoska, A. Gradišek, A. M. Peterlin, M. Gams, and G. Poglajen, "Chronic heart failure detection from heart sounds using a stack of machine-learning classifiers," in Proceedings of the 13th International Conference on Intelligent Environments, pp. 14–19, 2017.
- B. Cvetković, M. Gjoreski, J. Šorn, M. Frešer, M. Luštrek, M. Bogdański, K. Jackowska, M. Kosiedowski, and A. Stroiński, "Management of physical, mental and environmental stress at the workplace," in Proceedings of the 13th International Conference on Intelligent Environments, pp. 76–83, 2017.
- B. Cvetković, M. Gjoreski, J. Šorn, P. Maslov, and M. Luštrek, "Monitoring physical activity and mental stress using wrist-worn device and a smartphone," in Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 414–418, 2017.
- B. Budna, M. Gjoreski, A. Gradišek, and M. Gams, "JSI Sound – Platforma za enostavno klasifikacijo zvočnih posnetkov: Demonstracija na zvokih živali," in Proceedings of the International Multiconference Information Society, pp. 23–26, 2017.
- M. Gjoreski, B. Mitrevski, M. Luštrek, and M. Gams, "R-R vs GSR – an inter-domain study for arousal recognition," in Proceedings of the International Multiconference Information Society, 79–82, 2017.
- B. Cvetković, M. Gjoreski, J. Šorn, Jure, P. Pavel, M. Kosiedowski, M. Bogdański, A. Stroiński, and M. Luštrek, "Real-time physical activity and mental stress management with a wristband and a smartphone," in UbiComp/ISWC'17 Adjunct

Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 225–228, 2017.

Book Chapters

- H. Gjoreski, S. Stankoski, I. Kiprijanovska, A. Nikolovska, N. Mladenovska, M. Trajanoska, B. Velichkovska, M. Gjoreski, M. Luštrek, and M. Gams, “Wearable sensors data-fusion and machine-learning method for fall detection and activity recognition,” in *Challenges and Trends in Multimodal Fall Detection for Healthcare*, (Studies in Systems, Decision and Control, vol. 273). Cham: Springer, pp. 80–96, 2020.
- V. Janko, M. Gjoreski, G. Slapničar, M. Mlakar, N. Reščič, J. Bizjak, V. Drobnič, M. Marinko, N. Mlakar, M. Gams, and M. Luštrek, “Winning the Sussex-Huawei locomotion-transportation recognition challenge,” in *Human Activity Sensing: Corpus and Applications*, (Springer Series in Adaptive Environments). Cham: Springer, pp. 233–250, 2019.

Workshop Papers

- M. Gjoreski, A. Gradišek, B. Budna, M. Gams, and G. Poglajen, “Toward early detection and monitoring of chronic heart failure using heart sounds,” in *Workshop Proceedings of the 15th International Conference on Intelligent Environments*, pp. 336–343, 2019.
- V. Janko, M. Gjoreski, C. De Masi, N. Reščič, M. Luštrek, and M. Gams, “Cross-location transfer learning for the Sussex-Huawei locomotion recognition challenge,” in *UbiComp/ISWC'19 Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 730–735, 2019.
- M. Gjoreski, S. Kalabakov, M. Luštrek, M. Gams, and H. Gjoreski, “Cross-dataset deep transfer learning for activity recognition,” in *UbiComp/ISWC'19 Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 714–718, 2019.
- M. Gjoreski, M. Luštrek, and V. Pejović, “My watch says I'm busy: Inferring cognitive load with low-cost wearables,” in *UbiComp/ISWC'18 Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 1234–1240, 2018.
- M. Gjoreski, V. Janko, N. Reščič, M. Mlakar, M. Luštrek, J. Bizjak, G. Slapničar, M. Marinko, V. Drobnič, and M. Gams, “Applying multiple knowledge to Sussex-Huawei locomotion challenge,” in *UbiComp/ISWC'18 Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 1488–1496, 2018.
- V. Janko, N. Reščič, M. Mlakar, V. Drobnič, M. Gams, G. Slapničar, M. Gjoreski, J. Bizjak, M. Marinko, and M. Luštrek, “A new frontier for activity recognition : the Sussex-Huawei locomotion challenge,” in *UbiComp/ISWC'18 Adjunct Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 1511–1520, 2018.
- M. Gjoreski, M. Luštrek, and M. Gams, “Multi-task ensemble learning for affect recognition,” in *UbiComp/ISWC'19 Adjunct Proceedings of the ACM International joint Conference on Pervasive and Ubiquitous Computing*, pp. 1511–1520, 2018.
- M. Gjoreski, B. Budna, A. Gradišek, and M. Gams, “JSI Sound : a machine-learning tool in Orange for classification of diverse biosounds,” in *Workshop on Advances in Bioinformatics and Artificial Intelligence: Bridging the Gap*, co-located with 26th International Joint Conference on Artificial Intelligence, 2017.

- M. Gjoreski, H. Gjoreski, M. Luštrek, and M. Gams, “Deep affect recognition from R-R intervals,” in UbiComp/ISWC'17 Adjunct Proceedings of the 2017 ACM International joint Conference on Pervasive and Ubiquitous Computing, pp. 754–762, 2017.

Other Publications

- M. Gjoreski, S. Kochev, N. Reščič, M. Gregorič, T. Eftimov, and B. Koroušić-Seljak, “Exploring dietary intake data collected by FPQ using unsupervised learning,” in Proceedings of the International Conference on Big Data, pp. 5126–5130, 2019.
- R. Piltaver, M. Luštrek, S. Džeroski, M. Gjoreski, and M. Gams, “Learning comprehensible and accurate hybrid trees,” Expert Systems With Applications, vol. 164, p. 113980. **IF: 5.452, (A', 1/4).**

Biography

Martin Gjoreski was born on May 12, 1991 in Prilep, N. Macedonia. He finished his undergraduate studies in 2014 with great honor, at the Faculty of Computer Science and Engineering in Skopje. During his undergraduate studies he was awarded a scholarship by the Ministry of Education of N. Macedonia, which was given to the best 125 students with remarkable success in the country.

In 2014, he joined the Department of Intelligent Systems at the Jožef Stefan Institute as a student researcher, advised by Prof. Dr. Matjaž Gams and Dr. Mitja Luštrek. In 2016, he finished his master's studies at the Jožef Stefan International Postgraduate School. His Master's work in the field of monitoring stress using wearable devices was funded by the Slovene Human Resources Development and Scholarship Fund.

Since 2016, he is employed as a Young Researcher at the Department of Intelligent Systems at the Jožef Stefan Institute, under the advisement of Prof. Dr. Matjaž Gams and Dr. Mitja Luštrek. His research field is AI with focus on the development of standard and deep machine learning methods for sensor data. He is particularly interested in applications in fields such as ambient intelligence, mobile computing and mobile health. His PhD work is funded by the Slovenian Research Agency.

Since 2019, he is also a teaching assistant at the Faculty of Computer and Information Science, University of Ljubljana, teaching "Platform Based Development" at a Bachelor's level and "Mobile Sensing" at a Master's level.

His research work has been presented at dozens of international conferences and published in several international journals. He was also part of the teams that won five machine-learning challenges organized at A-ranked conferences (UbiComp, IJCNN and ABC Conference): SHL 2018, SHL 2019, EmteqAR 2019, Challenge-UP 2019 and Cooking AR Challenge 2020.