CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Improving human movement sensing with micro models and domain knowledge
Authors	Scheurer, Sebastian
Publication date	2021-06-18
Original Citation	Scheurer, S. 2021. Improving human movement sensing with micro models and domain knowledge. PhD Thesis, University College Cork.
Type of publication	Doctoral thesis
Rights	© 2021, Sebastian Scheurer. - https://creativecommons.org/licenses/by-nc-nd/4.0/
Download date	2026-11-16 11:36:45
Item downloaded from	https://hdl.handle.net/10468/11868



UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Improving Human Movement Sensing with Micro Models and Domain Knowledge

Sebastian Scheurer

MSc.
114221860

Thesis submitted for the degree of Doctor of Philosophy



NATIONAL UNIVERSITY OF IRELAND, CORK

College of Science, Engineering and Food Science
School of Computer Science & Information Technology
Insight Centre for Data Analytics

June 2021

Head of Department: Prof. Cormac J. Sreenan

Supervisors: Prof. Ken Brown
Prof. Barry O'Sullivan

Contents

List of Figures iv
List of Tables v
Abstract viii
Acknowledgements x

1 Introduction **1**
1.1 Motivation 1
1.2 Thesis Statement 5
1.3 Contributions 6
1.3.1 Wearable Human Activity Recognition for Emergency First Responders 6
1.3.2 Subject-Dependent and -Independent Human Activity Recognition with Micro and Macro Models 6
1.3.3 Expert Hierarchies for Human Activity Recognition . . 7
1.3.4 Device-Free Human Movement Detection from Ambient Wi-Fi Signal Data with Micro Models 7
1.4 Publications 8
1.5 Thesis Structure 10

2 Related Work **12**
2.1 Human Activity Recognition from Wearable Inertial Sensor Data 12
2.1.1 The State of the Art in Wearable Human Activity Recognition 13
2.1.2 Subject-Dependent and Subject-Independent Human Activity Recognition 16
2.1.3 Personalising Human Activity Recognition Models . . . 19
2.1.4 Multi-Class Decomposition Methods for Human Activity Recognition 22
2.2 Device-Free Human Presence Detection from Wi-Fi Signal Data 28
2.2.1 Methods Based on Received Signal Strength 30
2.2.2 Methods Based on Channel State Information 31
2.3 Conclusions 40

3 Wearable Human Activity Recognition for Emergency First Responders 43
3.1 Methods . . . 46
3.1.1 Experimental Design and Data Acquisition . . . 46
3.1.2 Data Pre-Processing . . . 50
3.1.3 Feature Extraction . . . 51
3.1.4 Algorithm Tuning . . . 53
3.1.5 Sensor and Feature Selection . . . 54
3.2 Results and Discussion . . . 55
3.2.1 Sensor and Feature Selection . . . 58
3.3 Conclusions . . . 63

4 Subject-Dependent and -Independent Human Activity Recognition with Micro and Macro Models 65
4.1 Methods . . . 67
4.1.1 Pre-Processing, Segmentation, and Feature Extraction 71
4.1.2 Activity Inference and Evaluation . . . 72
4.2 Results and Analysis . . . 73
4.2.1 Analysis of the Subject-Dependent Performance . . . 78
4.2.2 Analysis of the Subject-Independent Performance . . . 82
4.2.3 Comparing Subject-Dependent and Subject-Independent Performance . . . 85
4.3 Discussion . . . 86
4.4 Conclusions . . . 90

5 Human Activity Recognition with Expert Hierarchies 92
5.1 Multi-Class Decomposition Methods . . . 95
5.1.1 Flat Decomposition Strategies . . . 95
5.1.2 Error-Correcting Output Codes . . . 96
5.1.3 Hierarchical Decomposition Strategies—Nested Dichotomies . . . 97
5.2 A Shortcut to Discrete Predictions for Nested Dichotomies . . 99
5.3 Computational Experiments and Evaluation . . . 100
5.4 Results . . . 103
5.5 Discussion . . . 110
5.6 Conclusions . . . 117

6 Detecting Human Movement from Ambient Wi-Fi Signal Strength with Micro Models 119
6.1 Wi-Fi Testbeds . . . 122

6.2 Acquiring Wi-Fi Signals for Evaluating Human Presence Detection Methods . . . 124
6.3 Human Movement Detection and its Evaluation . . . 127
6.4 Results and Discussion . . . 129
6.5 Conclusions . . . 138

7 Conclusions & Future Work **140**
7.1 Conclusions . . . 141
7.1.1 Human Activity Recognition for Emergency First Responders . . . 141
7.1.2 Subject-Dependent and -Independent Human Activity Recognition with Micro and Macro Models . . . 141
7.1.3 Human Activity Recognition with Expert Hierarchies . 143
7.1.4 Detecting Human Movement from Ambient Wi-Fi Signal Strength with Micro Models . . . 144
7.2 Future Work . . . 146
7.2.1 Micro and Macro Models . . . 146
7.2.2 Expert Hierarchies . . . 147
7.2.3 Device-free Human Sensing with Wi-Fi Signals . . . 149

Acronyms **151**

A Additional Performance Metrics for Micro and Macro Models **174**

B Expert Hierarchies **181**

List of Figures

3.1 Firefighters engaging in some of the activities of interest . . . 47
3.2 Inertial Measurement Unit 48
3.3 Inertial signals and assorted features for three activity examples 52
3.4 Number of features per K-W percentile and sensor 62

4.1 Number of instances per activity for each data-set 69
4.2 Graphical summary of the experiment 71
4.3 Subject-independent vs subject-dependent κ across learning algorithms 74
4.4 Distribution of user-wise subject-dependent ranks across learning algorithms and personalisation-generalisation approaches . 81
4.5 Estimated marginal means for subject-dependent performance across learning algorithms and personalisation-generalisation approaches 84
4.6 Distribution of user-wise subject-independent ranks across learning algorithms and personalisation-generalisation approaches 85
4.7 Estimated marginal means for subject-dependent and -independent performance across learning algorithms and personalisation-generalisation approaches 87

5.1 C.I.s for effect of multi-class decomposition method on performance on the 17-class problem 107
5.2 C.I.s for effect of multi-class decomposition method on performance on the EH1 and EH4 problems 108

6.1 Schematic of the four rooms, drawn to scale, where we performed our experiments 123
6.2 ROC curves of bagged single-link model predictions 132

B.1 Expert hierarchy 1 (EH1) 181
B.2 Expert hierarchy 2 (EH2) 182
B.3 Expert hierarchy 3 (EH3) 183
B.4 Expert hierarchy 4 (EH4) 184
B.5 Expert hierarchy 5 (EH5) 185

List of Tables

3.1 MAE and overall Accuracy (%): 17-activities 56
3.2 MAE and overall Accuracy (%): move-type/lie 57
3.3 MAE and overall Accuracy (%): Move-type 57
3.4 MAE (± SE) and SD (all in %) for all Sensor combinations . . 59
3.5 MAE (± SE) and SD (all in %) when using PCA 59
3.6 MAE (± SE) and SD (all in %) with K-W feature selection . . 59
3.7 MAE (± SE) for GBT with K-W, retaining different percentiles 61

4.1 Description and summary statistics of the HAR data-sets . . . 68
4.2 Subject-dependent and -independent κ across learning algorithms and personalisation-generalisation approaches 75
4.3 GLMM analysis of the subject-dependent performance across learning algorithms and personalisation-generalisation approaches 80
4.4 GLMM analysis of subject-independent performance across learning algorithms and personalisation-generalisation approaches 83
4.5 GLMM analysis of the subject-dependent and -independent performance across personalisation-generalisation approaches . 87

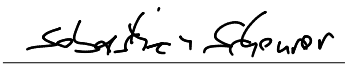
5.1 Mean κ of different multi-class decomposition methods and learning algorithms on the multi-class HAR problem 104
5.2 Mean κ of different multi-class decomposition methods and learning algorithms on the binary EH1 problem 105
5.3 Mean κ of different multi-class decomposition methods and learning algorithms on the binary EH4 problem 106
5.4 Estimated logistic regression coefficients for the multi-class problem 111
5.5 Estimated logistic regression coefficients for the binary EH1 problem 112
5.6 Estimated logistic regression coefficients for the binary EH4 problem 113

6.1 Median accuracy (%) when models trained on single links from the training room are tested on single links from the test room 130
6.2 Median accuracy (%) across single-link models when averaging test-link predictions within each 30 s window 131
6.3 Median accuracy (%) when models trained on all links from the training room are applied to single test links 133

6.4 Accuracy (%) when averaging the test-link predictions of models trained on all links from the training room 134
6.5 Accuracy (%) when models are trained with the best or all the links from the other rooms, and tested on the average (median) link or by averaging all test-link predictions 136
6.6 Median accuracy (%) when R-TTWD is trained with single links from the training room and tested on single links from the test room 137
6.7 Median accuracy (%) when averaging the single-link R-TTWD predictions prior to combining the three CSI streams via majority vote 137

A.1 Subject-dependent and -independent accuracy across learning algorithms and personalisation-generalisation approaches . . . 175
A.2 Subject-dependent and -independent weighted F1-score across learning algorithms and personalisation-generalisation approaches 178

I, Sebastian Scheurer, certify that the work I am submitting is my own and has not been submitted for another degree, either at University College Cork or elsewhere. All external references and sources are clearly acknowledged and identified within the contents. I have read and understood the regulations of University College Cork concerning plagiarism.



Sebastian Scheurer

Abstract

Human sensing is concerned with techniques for inferring information about humans from various sensing modalities. Examples of human sensing applications include human activity (or action) recognition, emotion recognition, tracking and localisation, identification, presence and motion detection, occupancy estimation, gesture recognition, and breath rate estimation.

The first question addressed in this thesis is whether micro or macro models are a better design choice for human sensing systems. Micro models are models exclusively trained with data from a single entity, such as a Wi-Fi link, user, or other identifiable data-generating component. We consider micro and macro models in two human sensing applications, viz. human activity recognition (HAR) from wearable inertial sensor data and device-free human presence detection from Wi-Fi signal data. The HAR literature is dominated by person-independent macro models. The few empirical studies that consider both micro and macro models evaluate them with either only one data-set or only one HAR algorithm, and report contradictory results. The device-free sensing literature is dominated by link-specific micro models, and the few papers that do use macro models do not evaluate their micro counterparts. Given the little and contradictory evidence, it remains an open question whether micro or macro models are a better design choice. We evaluate person-specific micro and person-independent macro models across seven HAR benchmark data-sets and four learning algorithms. We show that person-specific models (PSMs) significantly outperform the corresponding person-independent model (PIM) when evaluated with known users. To apply PSMs to data from new users, we propose ensembles of PSMs, which are improved by weighting their constituent PSMs according to their performance on other training users. We propose link-specific micro models to detect human presence from ambient Wi-Fi signal data. We select a link-specific model from the available training links, and show that this approach outperforms multi-link macro models.

The second question addressed in this thesis is whether human sensing methods can be improved with domain knowledge. Specifically, we propose

expert hierarchies (EHs) as an intuitive way to encode domain knowledge and simplify multi-class HAR, without negatively affecting predictive performance. The advantages of EHs are that they have lower time complexity than domain-agnostic methods and that their constituent classifiers are statistically independent. This property enables targeted tuning, and modular and iterative development of increasingly fine-grained HAR. Although this has inspired several uses of domain-specific hierarchical classification for HAR applications, these have been ad-hoc and without comparison to standard domain-agnostic methods. Therefore, it remains unclear whether they carry a penalty on predictive performance. We design five EHs and compare them to the best-known domain-agnostic methods. Our results show that EHs indeed can compete with more popular multi-class classification methods, both on the original multi-class problem and on the EHs' topmost levels.

Acknowledgements

If there is one person without whom this thesis would not have come into being, then it is my supervisor Prof. Ken Brown. Your open mind and ears, patience and optimism, encouragement, confidence and trust, sense of humour, and experience and expertise have been the foundation on which I carried out the research which has now crystallised into this thesis, and I am grateful for it. Thank you.

I further would like to thank the wonderful administrative and technical support staff of the Insight Centre for Data Analytics at University College Cork. Caitríona Walsh, Linda O'Sullivan, and Eleanor O'Riordan, without whose tireless administrative work nothing in the Insight Lab would function. Special thanks also goes to Peter J. McHale for his assistance with the acquisition and setup of the many devices involved in our Wi-Fi experiments, and to the school of computer science's IT services team, Martin Fleming and David O'Byrne, who were always ready to assist, troubleshoot, and resolve technical issues. Furthermore, our summer intern Tim Creedon deserves special mention for setting up and testing the Wi-Fi testbeds, and writing code to automate the Wi-Fi experiments and process the resulting data. I have no doubt that completing this part of my research in a timely fashion would have been impossible without your skills, commitment, and perseverance.

I have no doubts that assisting Dr. Marc van Dongen in delivering the postgraduate "Data Mining" module over the course of several semesters is what prepared me for the experience of teaching that module by myself. My thanks go to him and to Prof. Cormac Sreenan for giving me the opportunity to teach that module as my first own module.

I cannot pass the opportunity to thank all the wonderful people who are the Insight Centre for Data Analytics, particularly my colleagues in the School of Computer Science and Information Technology at the University College Cork. It is them who make the Insight Centre what it is, a place in which everyone can work, contribute, and grow in the manner that best suits them, without having to sacrifice their personal life. I also want to express my thanks to my examiners, Dr. Derek Bridge and Prof. Tomas Ward, for their

expertise, time, and engagement with my thesis. I could not have wished for a better pair of minds to examine and discuss my work.

Furthermore, I want to express my gratitude to Dr. Brendan O'Flynn and his wireless sensor networks research group at the Tyndall National Institute for contributing their time, facilities, equipment, and expertise, all of which were instrumental to my human activity recognition experiments. These experiments would not have been possible without the people who volunteered their time, energy, and quite a lot of sweat to walk, run, crawl, fall, and jump around the facilities with heavy backpacks and boots. Although I cannot name you without forfeiting your privacy, rest assured that I sincerely appreciate your contribution. I am particularly grateful to the first few volunteers' patience while we addressed issues with the hardware platform, and refined the data acquisition protocols and the software tools that supported them. I particularly want to thank Salvatore Tedesco for his support with organising and running the experiments, and many interesting conversations about human sensing, all of which were critical to getting the papers which provide much of the material in this thesis written, submitted, and published. Further thanks go to Blanca Florentino-Liaño and Piyush Agrawal whose eyes, experience, and ideas inspired many refinements and improvements in experimental design and implementation, and analysis of the results.

Finally, I want to express my deepest gratitude to my beloved partner, Sylwia, and my children, Moyamira Eva Jasmine, Síofra Adama Callí, and Emil Finnbar. It is you who inspire and motivate me to strive to listen better and judge less, be more empathetic and open-minded, and to never ever give up, but to try again (and fail better). I could not, and would not want to, imagine the journey that lead to this thesis without you. I also want to take this opportunity to thank my parents, Kathrin and Ändsg, not only for their financial support when the going got tough and without which I could not have gotten through what turned out to be over a decade of higher education, but most of all for instilling me at a young age with curiosity, optimism, and a habit for learning, reading, and critical thinking.

This thesis was supported by funding from Science Foundation Ireland (SFI) under grant number 12/RC/2289-P2 which is co-funded under the

European Regional Development Fund, a collaborative research grant from United Technologies Research Centre, the European Regional Development Fund under grant number 13/RC/2077-CONNECT, and the European-funded project SAFESENS under the ENIAC program in association with Enterprise Ireland (EI) under grant number IR20140024.

Chapter 1

Introduction

1.1 Motivation

Human sensing is concerned with techniques for inferring information about humans from various sensing modalities. Examples of human sensing applications include human activity (or action) recognition, emotion recognition, tracking and localisation, identification, presence and motion detection, occupancy estimation, gesture recognition, and breath rate estimation. Human sensing systems have become an increasingly popular area for machine learning across a range of applications, including medical (e.g., monitoring patients), industrial (e.g., monitoring workers for movements with increased risk of repetitive strain injury), and home care and assisted living (e.g., monitoring the elderly for dangerous falls or signs of depression).

Human sensing systems can broadly be grouped into two categories, according to whether the sensors are placed on (or in) the user's body or in the environment. The former is known as wearable sensing and the latter as "sensor-less," "device-free," or "environmental" sensing. Some applications are more amenable to wearable than to device-free sensing, and vice versa. In applications, such as presence detection, that refer to a specific area of interest, such as a room or elevator, it is often hard and sometimes impossible to design a system that only relies on wearable sensing modalities. Instead, it is more natural to deploy an environmental sensing system that directly

(and only) monitors the area of interest, be it via cameras, ambient radio signals, or thermal imaging. In other applications, such as human activity recognition and breath rate estimation, both wearable and environmental sensing solutions are plausible.

Despite their differences, human sensing systems also have their commonalities. For one, they invariably can be viewed as prediction problems. Tasks such as human activity recognition (HAR), presence and movement detection, and gesture recognition naturally lend themselves to formulation as classification problems, whereas others such as breath rate or occupancy estimation are more naturally viewed as regression problems. The prediction problem formulation makes them amenable to the whole battery of predictive statistical and machine learning algorithms that have garnered much attention and seen much progress in recent years. With so much success, that it—applying machine learning algorithms to sensor data to make inferences about people and their behaviours—has become the predominant approach in the human sensing literature, and is also the approach taken in this thesis.

Human sensing systems can also be categorised on whether they use micro- or macro-models. Micro models, as opposed to macro models, are models exclusively trained with data from a single entity in the human sensing system. Such an entity can correspond to many things—an individual sensor node, a Wi-Fi link, a user, or any other identifiable component that generates data which feed the inference algorithm. Conversely, macro models are trained without giving any consideration to which entity generated a given instance's data. Micro models (albeit under different names) appear in the human sensing literature in the context of subject-dependent and -independent performance of HAR inference algorithms. Wearable HAR systems are customarily evaluated for their ability to either generalise to unknown users (people not represented in the HAR algorithm's training data) or to known users (people represented in the training data), with the former known as *subject-independent* and the latter as *subject-dependent* performance. Which of the two performance types should be optimised depends on how the system is going to be commissioned and deployed. If commissioning the system entails obtaining examples of the activities of interest from its

end users—the people whose activities the deployed system has to recognise—then we should optimise the subject-dependent performance. If, on the other hand, the system is to be deployed without prior commissioning (i.e., without being trained on data from its end users), then we should optimise the subject-independent performance. The subject-dependent and -independent performance is usually estimated via k-fold cross-validation (CV) and leave-one-subject-out cross-validation (LOSO CV), respectively, across all users in the data-set. This evaluation implies that there is a single model that is shared by all users. This is an example of a macro model. Throughout this thesis, we refer to a macro model that is built with data from an entire user population as a person-independent model (PIM). While PIMs are the predominant approach in the literature, there are a few papers [BI04; WL12; BG17; Fer+20] that estimate the subject-dependent performance by performing a separate evaluation within each user's data, which implies that every user has their own individual model. These models are an example of micro models from the domain of wearable HAR. Throughout this thesis we refer to micro models such as these, which are built with and applied to data from individual users, as person-specific models (PSMs).

Micro models also appear in the context of device-free human sensing from radio frequency (RF) signals, with each model corresponding to an individual antenna or wireless link. Particularly methods that exploit Wi-Fi signals for human sensing applications, as we do in this thesis, tend toward micro models. Most of the existing methods rely on a handful of links, and all but a few [Wu+15; Qia+18] use micro models, which are usually built and tested with data from one link. Only a few papers [Li+17; Zhu+17] test their methods across links and facilities/rooms, and only one [Li+17] considers other changes in the environment, such as changes in the placement of furniture or wireless transceivers. Finally, most of the proposed device-free sensing systems rely on dedicated Wi-Fi nodes or at least on knowing the transceivers (relative) locations. To deploy such a system one has to survey each target environment, decide where to place each transceiver, and physically deploy and maintain them throughout the system's lifetime. In their reliance on dedicated Wi-Fi nodes, these approaches fail to exploit the ubiquity of Wi-Fi networks, and

are at odds with the real-world requirements of deploying device-free sensing systems at scale.

Although micro models do appear in the human sensing literature, the evidence whether, and how much, better they perform than their macro counterparts is inconclusive at best. Only a few papers [WL12; BG17; Fer+20] consider how the subject-dependent HAR performance of micro models (PSMs) compares to that of the corresponding macro model (PIM), and their results are contradictory. Furthermore, all of these papers include either only one data-set or only one learning algorithm in their evaluation. There is even less evidence on the subject that relates to device-free sensing, or at least to human presence detection from Wi-Fi signals. There are only two pertinent papers [Wu+15; Qia+18] that use a macro model, trained with data from multiple wireless links, for device-free human presence detection from Wi-Fi signal data, but no one has published a direct comparison between micro and macro models in this area. Given the little and conflicting evidence in the literature, it thus remains an open question whether micro or macro models are a better design choice for human sensing systems. This is one of the two questions addressed in this thesis.

The second question pertains to the use of domain knowledge in human sensing. Specifically, we consider domain knowledge in the form of activity hierarchies to decompose multi-class HAR problems into a set of binary classification problems, each of which can be tackled by a binary classifier. There are several methods to decompose multi-class problems into a set of binary classification problems, which can be grouped into flat and hierarchical decomposition methods. What most of them, particularly the flat ones, have in common is that the binary classifiers they generate are not statistically independent, and therefore cannot be analysed and tuned in isolation. Nested dichotomies have a long history in statistics as a standard tool for analysing polychotomous data with binomial logistic regression models [Fox97], and have the nice property that their constituent binary classifiers are mutually independent. This independence means that each classifier can be analysed and optimised independently.

They, and other hierarchical classification methods, are particularly at-

tractive for HAR because it is almost always natural and easy to arrange the activities of interest in a hierarchy. For example by placing the most general categories (e.g., "mobile" and "stationary") at the top or root of the tree, proceeding to increasingly specific categories ("walk" and "run"), and terminating with the most specific categories ("walk upstairs" and "walk downstairs") at the leaves. This enables the iterative and modular development, and improvement, of increasingly fine-grained HAR capabilities. Besides their conceptual simplicity, nested dichotomies also have lower space and time complexity, both at training and evaluation (prediction) time. These advantages have inspired several HAR applications of hierarchical classification [Mat+04; Kar+06; FG15; CY18], but there is a lack of research into how hierarchical methods compare to other decomposition methods. Thus, the question whether these advantages come with a price is an open one, and it remains unclear whether hierarchical classification can compete with other multi-class HAR methods. This is the second question addressed in this thesis.

1.2 Thesis Statement

Micro models and domain knowledge can improve human sensing methods. In particular:

- person-specific micro models can improve the subject-dependent performance of methods for HAR from wearable sensor data;
- link-specific micro models, either individually or in ensemble, can improve the predictive performance of device-free wireless human presence detection methods;
- domain knowledge encoded in expert hierarchies simplifies multi-class HAR models while maintaining predictive performance.

1.3 Contributions

To address our main goal—answer the question whether micro models and domain knowledge can improve human sensing methods—we make several contributions. They are described in this section.

1.3.1 Wearable Human Activity Recognition for Emergency First Responders

Most of the HAR literature is devoted to activities of daily living, and only few papers [Fra+14; AFH15] explore HAR for the dynamic activities and environments that are typical of emergency first response operations, with results that are clearly worse than the state-of-the-art in HAR from wearable inertial sensor data. In chapter 3, we develop a gradient boosted ensemble of decision trees (GBT) designed for human activity recognition for emergency first responders from a single inertial measurement unit (IMU). In experiments with a data-set covering 17 activities that are relevant to emergency first response team leaders, we compare our GBT with three other popular HAR algorithms across four different HAR problems. Our results show that our GBT clearly outperforms the other algorithms on three of these classification problems.

1.3.2 Subject-Dependent and -Independent Human Activity Recognition with Micro and Macro Models

In chapter 4, we assess micro models, i.e., PSMs, as a way to boost predictive HAR performance for users that are represented in the training data. We also propose three different flavours of ensembles of person-specific models (EPSMs) as alternatives to the macro PIM approach to subject-independent HAR that dominates the literature. We empirically compare the subject-dependent an -independent performance of PIMs, PSMs, and EPSMs on seven HAR benchmark data-sets. Our results show that PSMs indeed outperform the corresponding PIM in terms of subject-dependent performance by about

as much as a PIM outperforms the corresponding EPSM in terms of subject-independent performance. While EPSMs do not quite achieve the same subject-independent performance as the corresponding PIM, the κ-weighted EPSM comes within a few percentage points. EPSMs have the advantage that they can easily exploit data from a new user by fitting a PSM to the user's data and adding it to the ensemble, without accessing other users' data.

1.3.3 Expert Hierarchies for Human Activity Recognition

In chapter 5, we investigate the use of hierarchical classifiers (more commonly known in the statistical literature as "nested dichotomies" [Fox97]) in a HAR context. We present the first direct comparison of hierarchical classification that is guided by domain knowledge—an approach we term *expert hierarchies*—with standard domain-agnostic multi-class decomposition methods on a multi-class HAR problem. We formulate a novel threshold that indicates when a nested dichotomy's branch cannot possibly be on the path to the class with the largest predicted probability, and therefore does not need to be explored. We show that domain knowledge can be used to construct a multi-class HAR classifier that has lower computational complexity and is easier to interpret than, but performs comparably to the most commonly used multi-class decomposition method.

1.3.4 Device-Free Human Movement Detection from Ambient Wi-Fi Signal Data with Micro Models

In chapter 6, we present a method that can reliably detect the presence of a moving human occupant in a room using low cost off-the-shelf Wi-Fi transceivers. This is the first device-free sensing method that is based on the antenna-wise received signal strength (RSS), as opposed to the total received signal strength indicator (RSSI) or channel state information (CSI), from Wi-Fi networks. It makes no assumptions about Wi-Fi network topology or geometry, beyond there being a Wi-Fi access point (AP) with multiple

antennae in the monitored room and at least one transmitting client. We build micro models, corresponding to individual AP-client links, and evaluate them with multiple links in held-out test rooms. By opportunistically taking advantage of whatever clients are transmitting to the AP, we significantly improve accuracy compared to predictions that are based on an arbitrary single AP-client link.

We show that the single-link (micro) model with the best performance for the links in the training rooms also outperforms most multi-link (macro) models in the test room. In a like-for-like comparison with our implementation of R-TTWD [Zhu+17], a CSI-based state-of-the-art method for human presence detection, our approach performs comparably or better than R-TTWD when tested in larger rooms, and significantly better in smaller rooms. To foster further research in device-free human presence detection and activity recognition with Wi-Fi signals, we make our data-set publicly available. There are a few data-sets of Wi-Fi signals for human activity or gesture recognition [Guo+17; Zhe+19], all of which are devoid of data corresponding to human absence. Our data-set is the first publicly available data-set of Wi-Fi signals for human presence detection.

1.4 Publications

Most of the work presented in this thesis has been published in international peer-reviewed scientific journals and conferences. They are listed here.

1. Sebastian Scheurer, Salvatore Tedesco, Kenneth N. Brown, and Brendan O'Flynn. "Human Activity Recognition for Emergency First Responders via Body-Worn Inertial Sensors". In: *International Conference on Wearable and Implantable Body Sensor Networks* (Eindhoven, NLD, May 9–12, 2017). BSN. IEEE, May 2017. DOI: 10.1109/BSN.2017.7935994.

2. Sebastian Scheurer, Salvatore Tedesco, Kenneth N. Brown, and Brendan O'Flynn. "Sensor and Feature Selection for An Emergency First Responders Activity Recognition System". In: *Sensors* (Glasgow, GBR,

Oct. 29–Nov. 1, 2017). IEEE, Oct. 2017. DOI: 10.1109/ICSENS.2017.8234090.

3. Sebastian Scheurer, Salvatore Tedesco, Òscar Manzano, Kenneth N. Brown, and Brendan O'Flynn. "Monitoring Emergency First Responders' Activities via Gradient Boosting and Inertial Sensor Data". In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases* (Dublin, IRL, Sept. 10–14, 2018). ECML/PKDD. 2018. DOI: 10.1007/978-3-030-10997-4_53.

4. Sebastian Scheurer, Salvatore Tedesco, Brendan O'Flynn, and Kenneth N. Brown. "Comparing Person-Specific and -Independent Models on Subject-Dependent and -Independent Human Activity Recognition Performance". In: *Sensors* 20.13 (June 2020): *Sensor-Based Activity Recognition and Interaction.* ISSN: 1424-8220. DOI: 10.3390/s20133647, which is an extended version of

5. Sebastian Scheurer, Salvatore Tedesco, Kenneth N. Brown, and Brendan O'Flynn. "Subject-Dependent and -Independent Human Activity Recognition with Person-Specific and -Independent Models". In: *International Workshop on Sensor-based Activity Recognition and Interaction* (Rostock, DEU, Sept. 16–17, 2019). iWOAR. ACM, Sept. 2019. DOI: 10.1145/3361684.3361689.

6. Sebastian Scheurer, Salvatore Tedesco, Kenneth N. Brown, and Brendan O'Flynn. "Using Domain Knowledge for Interpretable and Competitive Multi-class Human Activity Recognition". In: *Sensors* 20.4 (Feb. 2020): *Inertial Sensors for Activity Recognition and Classification.* ISSN: 1424-8220. DOI: 10.3390/s20041208.

The data-set which we acquired, and use, to validate the methods presented in chapter 6 has been deposited with Zenodo, an online public repository for scientific artefacts [SCB20].

In addition to these works, we have also co-authored the following publication, which is only tangentially related to this thesis. Salvatore Tedesco

et al. “Motion Sensors-based Machine Learning Approach for the Identification of Anterior Cruciate Ligament Gait Patterns in On-the-Field Activities in Rugby Players”. In: *Sensors* 20.11 (May 2020). ISSN: 1424-8220. DOI: 10.3390/s20113029 [Ted+20].

1.5 Thesis Structure

This thesis is structured as follows:

- In Chapter 1, we have discussed the motivation behind our work and its main goal, and have summarised the contributions that emerged from it.
- In Chapter 2, we review the pertinent literature and further discuss its shortcomings.
- In Chapter 3, we present an initial application example of our work on HAR with wearable IMU data. In it, we highlight the main issues that motivate much of the work in subsequent chapters, and develop the core machine learning methods used in them.
- In Chapter 4, we address the question whether HAR micro models, which are personalised to a specific user, can boost the subject-dependent performance beyond that achieved with the standard macro model approach. We compare person-specific micro models (PSMs) and person-independent macro models (PIMs) across seven HAR data-sets and four learning algorithms, including the ones we developed in chapter 3. Our results and analyses show that the micro PSMs significantly outperform its macro counterpart, the PIM, on subject-dependent performance.
- In Chapter 5, we propose the use of domain knowledge in the form of expert hierarchies to simplify multi-class HAR models. Our evaluation, which includes many popular approaches for multi-class (HAR) classification, shows that expert hierarchies perform comparable or better than its most popular contender.

- In Chapter 6, we propose link-specific micro models to detect the presence of a moving human occupant in a room using signals from Wi-Fi networks running on low cost off-the-shelf components. We evaluate micro models built from a single link and macro models built from multiple links with data from unseen links and rooms. Our results show that while macro models tend to perform better than an arbitrary micro model, we can exploit the multitude of links, and thus micro models, to find a micro model that consistently and clearly outperforms macro models when evaluated with Wi-Fi signals from a different room.
- In Chapter 7, we end the thesis with a summary of our main conclusions and a discussion of directions for future work that would advance the thesis topics.

Chapter 2

Related Work

2.1 Human Activity Recognition from Wearable Inertial Sensor Data

Human activity recognition (HAR) from inertial sensor data has been a fruitful line of enquiry for over a decade. A particularly popular approach, which has proven successful in numerous HAR applications, is to extract a set of features from inertial data along a sliding window, and use the resulting matrix—whose rows and columns correspond to windows and features, respectively—as inputs to the machine learning algorithm [LL13; BBS14].

We begin our survey of the wearable HAR literature with a summary of the state-of-the-art in subsection 2.1.1. Due to the many benchmark data-sets, this part of our literature review is limited to results that relate to the seven publicly available benchmark data-sets that we also use in the experiments presented in chapter 4. Then, in subsection 2.1.2, we turn our attention to publications that investigate the subject-dependent and -independent HAR performance of deep and traditional learning algorithms. This is where we first encounter the micro and macro models which we call person-specific models (PSMs) and person-independent models (PIMs), respectively, in a HAR context. This topic is further explored in subsection 2.1.3, with a review of methods that use a limited amount of data from new users, called the support set, to build personalised models from other users' training data.

Finally, in subsection 2.1.4, we discuss the literature about the relative merit of different multi-class decomposition methods, with a particular focus on hierarchical classification and HAR.

2.1.1 The State of the Art in Wearable Human Activity Recognition

In a survey of 56 papers on deep learning—deep neural, convolutional, and recurrent neural networks, auto-encoders, and restricted Boltzmann machines—for sensor-based human activity recognition, Wang, Chen, Hao, Peng, and Hu [Wan+19] concluded in 2019 that there is no single "model that outperforms all others in all situations." Comparing the results from the original studies for three HAR data-sets, among them the Opportunity [Cha+13] data-set employed in chapter 4, their survey identifies four papers [JY15; ZWL15; OR16; HHP16] as the state of the art. The next two paragraphs summarise, in chronological order, the results from these four papers with respect to the predictive performance on the HAR data-sets that feature in chapter 4 of this thesis.

The first of the four papers is the work by Jiang and Yin [JY15]. The deep convolutional neural network (DCNN) they proposed in 2015, termed DCNN+, recognises human activities from signal and activity images which are obtained by transforming the signals from a single inertial measurement unit (IMU) via the Discrete Fourier Transform or 2D Wavelets. To disambiguate between pairs of classes with confused predictions (i.e., classes with similarly large predicted probabilities) they employ binary Support Vector Machine (SVM) classifiers. The three considered models (DCNN, DCNN+, and SVM) all achieved a subject-dependent accuracy of >99% on the FUSION data-set [Sho+14]. The second paper, published in the same year by Zhang, Wu, and Luo [ZWL15], proposes a different deep neural network (DNN) for human activity recognition. This DNN recognises human activities from the raw signals acquired from a wearable IMU, and the signal magnitude of the accelerometer's combined three axes. In an empirical comparison, the authors pit their DNN against traditional (i.e., not deep learning) machine learning

algorithms with default, untuned, hyper-parameters that operate on five statistical features (mean, standard deviation, energy, spectral entropy, and pairwise correlations between the accelerometer axes) extracted from the raw IMU signals. Their results show their DNN achieving a subject-dependent error rate of 18% on the Opportunity data-set, SVM being a close runner-up with an error rate of 19%.

The third paper, published in 2016, is the work by Ordóñez and Roggen [OR16], in which they propose a deep convolutional long short-term memory (LSTM) model for human activity recognition. Their LSTM outperformed the baseline DCNN in terms of subject-dependent performance (F1-score: 93% vs. 91%) on the Opportunity data-set. The fourth paper was published in the same year by Hammerla, Halloran, and Plötz [HHP16]. In it, the authors compare DCNNs, DNNs, and three different types of LSTMs across three benchmark HAR data-sets, all of them consisting of data from multiple IMUs per user. Among them are two data-sets, the Opportunity and PAMAP2 [RS12] data-sets, which we also use in chapter 4 of this thesis. This paper is particularly elucidating because of its exploration of deep learning models' sensitivity to the many hyper-parameters that determine and control their architecture, learning, and parameter regularisation. The authors explore the search space by randomly sampling hyper-parameter configurations in hundreds and thousands of experiments. The results clearly show that deep neural networks are extremely sensitive to hyper-parameter settings, which is illustrated by the differences between each model's median and best performance. On the Opportunity data-set, the best model's (LSTM) median score is 17 percentage points lower than its best score, and on the PAMAP2 data-set the best model's (DCNN) median score is 7 percentage points lower than its best score. This latter number is the smallest discrepancy between best and median score across all models and data-sets. The best subject-dependent performance on the Opportunity data-set, an F1-score of 93%, was achieved with a bi-directional LSTM, whereas the best subject-dependent performance on the PAMAP2 data-set, an F1-score of 94%, was achieved with a DCNN. In their conclusions, the authors concur with Wang, Chen, Hao, Peng, and Hu in that no single model dominates across all data-sets.

In 2019, Abdu-Aguye and Gomaa [AG19] proposed an approach to feature extraction for sensor-based HAR. Their approach applies a wavelet transform and subjects the resulting decompositions to Spatial Pyramid Pooling [He+15] to obtain fixed-length features which preserve both local and global patterns of the input signals. They used these features as inputs to a random forest, and compared the performance against a DCNN. They estimated the subject-dependent performance by averaging over 15 repeated 75% train/25% test splits. Their method achieved a subject-dependent accuracy of 89% and an F1-score of 90%, while the DCNN achieved an accuracy of 87% and an F1-score of 87% on the REALWORLD data-set [SS16].

In 2020, Vakili, Ghamsari, and Rezaei [VGR20] evaluated seven machine learning algorithms, presumably operating on a set of extracted features, an artificial and a convolutional neural network, and a LSTM model on a range of internet-of-things data-sets. The data-sets' application domains range from occupancy detection from environmental sensors, to rain prediction from weather station data, to HAR from wearable IMUs. They found that random forests outperformed the other methods on the SIMFALL [ÖB14] data-set with an average subject-dependent accuracy of 76%, estimated via ten-fold cross-validation.

In 2018, Alharbi and Farrahi [AF18] proposed a DCNN to recognise smoking activities from a smartphone and smartwatch's inertial measurement units. They evaluated their approach on the UTSMOKE [Sho+16] data-set, but excluded the walking, sitting, and standing activities "because they were very simple to classify using the DCNN." They report an average subject-independent F1 score of 90%, estimated via a 70% train/15% validation/15% test split, but do not clarify how these splits are construed in a subject-independent manner.

According to the results presented by the literature discussed in this subsection, deep learning outperforms machine learning with handpicked features on multi-IMU data by over 6%. However, when it comes to HAR from a single IMU—which is more convenient for end users who have to remember to wear and charge the IMUs—they lead to a different conclusion. If we consider only the results for single-IMU scenarios, then deep learning performs

comparably, or only marginally better, than (traditional) machine learning with handpicked features. Furthermore, many papers which show that deep learning outperforms traditional machine learning by a large margin compare a deep architecture, carefully tailored and tuned to the data-set, against machine learning algorithms that use default hyper-parameters and operate on a handful of basic features, which may not be a fair comparison. It is, therefore, too early to altogether abandon machine learning with handpicked features for HAR applications. In chapter 4, we apply several machine learning algorithms to a set of handpicked features extracted from a single IMU from eight different HAR benchmark data-sets. The results from these experiments are further evidence that machine learning with handpicked features can (still) achieve state-of-the-art performance, particularly if we build and apply the user-specific micro models we call PSMs.

2.1.2 Subject-Dependent and Subject-Independent Human Activity Recognition

Bao and Intille [BI04] assess the subject-dependent performance of PSMs for recognising 20 activities of daily living (ADLs) across 20 users by training four learning algorithms on a set of semi-controlled laboratory data and evaluating them on a set of semi-naturalistic data, and the subject-independent performance of a PIM by performing a leave-one-subject-out cross-validation (LOSO CV) on the combined data from both sets. In a second experiment, they assess the subject-dependent performance of PSMs trained with laboratory data from three new users, and the subject-independent performance of a PIM trained with laboratory data from five different users, evaluating both PSMs and PIM with semi-naturalistic data from the three new users. Unfortunately, the differences in the protocols for estimating subject-dependent and -independent performance in the first experiment means that we cannot compare them directly (the latter accuracies are 18% to 50% *higher* than the former). Their second experiment, which affords a fairer comparison, directly contradicts these findings: the subject-dependent PSM accuracy (77%) exceeds the subject-independent PIM accuracy (73%) by 6%. Weiss and

Lockhart [WL12] assess the subject-independent and -dependent performance of PIMs, and the subject-dependent performance of PSMs for recognising six ADLs using eight learning algorithms and a single data-set with 59 users. They report that PSMs outperform the corresponding PIM by 2% to 30% on subject-dependent accuracy, and that a PIM achieves 11% to 41% higher subject-dependent than -independent accuracy.

In 2019, Jordao, Nazare, Sena, and Schwartz [Jor+19] evaluated seven state-of-the-art HAR methods published between 2010 and 2016, including the work by Jiang and Yin [JY15] which we discussed in subsection 2.1.1, on six publicly available HAR benchmark data-sets. The data-sets include the PAMAP2 and MHEALTH [Bañ+14] data-sets, which are also used in chapter 4 of this thesis. Three of the seven methods entirely rely on deep neural networks, while the remaining four use handpicked features as inputs to classification algorithms. Each method was evaluated with a different data segmentation strategy (overlapping and non-overlapping sliding windows), and its predictive performance estimated via stratified k-fold, leave-one-subject-out, and leave-trials-out cross-validation (CV) across all users.

Leave-trials-out CV, which is discussed in more detail in section 4.1, segments data into overlapping sliding windows one trial at a time. A trial corresponds to one individual's performance of one activity or sequence of activities during the data acquisition. The results show that naïve resampling methods, such as stratified k-fold CV, tend to inflate the predictive performance when used with overlapping sliding windows, because the overlap between subsequent windows can appear in both the training and test data. Based on the (unbiased) subject-dependent performance, which was estimated via leave-trials-out CV across all users, and the subject-independent performance, which was estimated via LOSO CV, the authors identify two methods as the state-of-the-art in HAR from wearable sensor data.

The first is an ensemble classifier, consisting of a decision tree, logistic regression, and a shallow multi-layer perceptron, proposed in 2015 by Catal, Tufekci, Pirmit, and Kocabag [Cat+15]. This ensemble achieved a subject-dependent accuracy of 92% and 81% on MHEALTH and PAMAP2, respectively, with an average of 76%, and a subject-independent accuracy of

95% and 85% on MHEALTH and PAMAP2, respectively, with an average of 69% across all six data-sets. The second method that emerged is the DCNN proposed in 2015 by Chen and Xue [CX15]. This DCNN achieved a subject-dependent accuracy of 90% and 82% on MHEALTH and PAMAP2, respectively, with an average of 83%, and a subject-independent accuracy of 89% and 83% on MHEALTH and PAMAP2, respectively, with an average of 78% across the five data-sets to which the authors were able to apply it. The 7 to 10 percentage points difference between the two methods' averages is largely due to the ensemble's poor performance on the data-set to which the DCNN could not be applied. If we only consider the five data-sets that both methods were applied to, then the two methods perform comparably. The ensemble outperforms the DCNN on three out of five data-sets in terms of subject-independent performance and the DCNN outperforms the ensemble on three out of five data-sets in terms of subject-dependent performance, and the discrepancies in each performance type are within the margin of error for all but one data-set. The authors' analysis concludes that, although deep neural networks such as DCNNs have achieved remarkable results in HAR from wearable sensor data, in many cases machine learning algorithms operating on handpicked features can achieve comparable results.

As these results show, there is a clear tendency for methods to achieve better subject-dependent than -independent performance. However, as evidenced by the results reported by Jordao, Nazare, Sena, and Schwartz for the PAMAP2 and MHEALTH data-sets, this is by no means a given, at least not when using the monolithic PIM. There is only one paper [WL12] that directly estimates the subject-dependent performances of both PSMs and PIMs in a way that affords meaningful comparison. It estimates that PSMs outperform the corresponding PIM by 2% to 30%, depending on the learning algorithm, on subject-dependent accuracy. There are, however, other results, such as those from the first experiment by Bao and Intille [BI04], that cast doubt on whether PSMs really are superior to a PIM, even when comparing the formers' subject-dependent against the latter's subject-independent performance. In chapter 4, we present experiments with eight HAR data-sets and four learning algorithms to settle this question. The results and analysis

presented there clearly show that the PIM approach is indeed more often than not outperformed by PSMs on subject-dependent performance.

2.1.3 Personalising Human Activity Recognition Models

We now turn our attention to the literature on personalising wearable HAR models. These personalisation methods typically operate in a scenario in which there are labelled training data from a reasonably large group of users, and a small amount of data (usually also labelled), called the support set, from each end user. Although these end users are sometimes also represented in the training data, it is more common to hold out all of their data for testing.

Brena and Garcia-Ceja [BG17] propose a method to build personalised HAR models from a user-specific training set. Their method uses a small amount of labelled data from the test user (i.e., the support set) to select a subset of similar instances among the training users' data, and combine them with the support set into a personalised training set. Instances are selected by clustering a random subset of each activity's instances, and selecting the cluster that contains most of the test user's support set. These personalised training data are then used to fit the user-specific model, a CART decision tree. They evaluate their approach on four publicly available HAR data-sets, while varying the size of the support set from 1% to 30%. For comparison, they also estimate not only the subject-independent performance of PIMs (which they call "general models"), but also the subject-dependent performance of PIMs and PSMs (which they call "user-dependent models") when given access to the support set during training.

They compare their personalised models to PSMs and PIMs via paired t-tests over the entirety of the considered percentage range. The personalised models achieve improvements of 16 to 21, 10 to 14, 5 to 7, and 2 to 6 percentage points (95% Confidence Intervals) when compared to the corresponding PIM's subject-independent performance. They do not compare against the subject-dependent PIM performance, because the added support set improves the

performance so little that it is nearly imperceptible when plotted against the performance of PSMs and personalised models. According to the paired t-tests, personalised models outperform PSMs by 0.6 to 4.8, 17 to 27, and 16 to 34 percentage points on three data-sets, but are outperformed by PSMs (by 3.1 to 5.9 points) on one data-set. Furthermore, personalised models do perform better than PSMs with support sets consisting of less than 3% of user data, but when the support set size is increased their performance increases at a much slower rate than the corresponding PSM's. Thus, by the time that the support consists of 3%, 9%, or 25% of the user's data—depending on the data-set—PSM performance matches or exceeds that of personalised models for all but one data-set. Unfortunately, the performances achieved by PSMs and PIM are only presented in graphical form, which makes it difficult to compare them quantitatively. What is clear is that PSMs do outperform the PIM, at least when, and even if, they are both trained with 30% support sets. It is also clear that the size of the gap between the two methods varies substantially between data-sets, ranging from what looks like less than one to over 25 percentage points. The same is also true for the assessed methods' accuracy. This approximately ranges from 40% on one data-set to nearly 90% on another for PIMs, and from 65% to about 95% for PSMs.

Sani, Wiratunga, Massie, and Cooper [San+17] propose a method similar to the one by Brena and Garcia-Ceja [BG17] insofar in that it builds personalised HAR models from user-specific training sets, which are a combination of the small user-provided support set and a subset of similar instances selected from the training users' data. The method differs in that it uses a nearest neighbours approach to determine which instances are most similar to the support set's medoid for each activity. They evaluate their approach with an SVM, performing a LOSO CV with a 30% support set on a single private data-set which covers nine activities and 50 users. While their evaluation does consider the subject-independent PIM performance, it does not include any subject-dependent performance results for comparison. Instead, they consider what happens when the number of neighbours selected for inclusion in the user-specific training set is increased from 10% to 100% of the training data. Their results show that their method clearly outperforms the (subject-

independent) PIM performance, particularly when the number of selected neighbours is less than 80% of the training data. The biggest improvement is achieved if just 30% of training data are selected, in which case the proposed method performs 5 percentage points better than the subject-independent PIM performance.

A year later, Sani, Wiratunga, Massie, and Cooper [San+18] presented a matching network that uses a support set of only 30 s (equivalent to six instances) per activity to build personalised HAR classifiers. They evaluate their approach on the data-set also used in their 2017 paper [San+17]. Their evaluation, which is done on a randomly selected eight-user holdout, includes k-Nearest Neighbours (kNN), SVM, and the same neural network as the one used in their matching network (minus the personalising embedding network). While PSMs are not considered in their evaluation, they do assess the (subject-dependent) performance of PIMs when their training data are augmented with the support set. This increases the F1-score of the best PIM (the one with an SVM) from 72.8% to 73.4%. Nevertheless, with an F1-score of 78.8% their matching network performs 5 and 6 percentage points better than the PIMs in terms of subject-dependent and -independent performance, respectively.

More recently, Ferrari, Micucci, Mobilio, and Napoletano [Fer+20] proposed to use the similarity between test and training users to weight the training instances during classifier training. They consider two different types of similarity. The first, termed "physical similarity," is simply the inverse (euclidean) distance between two users' demographic and personal features, such as age, gender, and body-mass index. The second type of similarity, termed "sensor similarity," is the inverse distance between the two users' predictive features (the ones extracted from the sensor data for feeding the HAR algorithms). They evaluate their approach with an Adaboost classifier on three publicly available HAR data-sets, two with 13 and one with 6 activities, from 22, 28, and 57 users, respectively. Although the authors state that they did experiment with SVM and kNN, they do not report these results because "the adoption of the similarity-based weighting procedure did not lead to remarkable accuracy modifications" with these classifiers.

With the Adaboost classifier, their similarity-weighted models outperform the PIM by 11 percentage points on subject-dependent performance with an average accuracy of 84.75%. However, they only achieve a 0.9 point improvement on the 70.19% accuracy subject-independent accuracy achieved with the PIM. They also assess the subject-dependent performance of PSMs, with surprisingly bad results (45.57% and 43.55% accuracy) for two data-sets, and a respectable 84.79% accuracy for the third. The authors explain that PSMs perform so badly because of differences in the segmentation process. This might be the case, since the only data-set on which PSMs perform well is also the only one that uses a peak-based segmentation approach with 3 s windows, whereas the two other data-sets employ a fixed 5 s sliding window with 0% and 5% overlap. Then again, there might be other factors at play such as the distribution of the activities of interest or the level of their inherent (dis-) similarity. Judging from Figure 6 in the paper, the data-set with good PSM performance appears to have a more balanced distribution of instances over the activities of interest, with the three most frequent activities (out of 13), constituting about 40% of the instances, being "Walking," "Running," and "Going Downstairs." It might be that the classifier confuses these activities less than "Standing" and "Sitting," which make up nearly half of the six-activity data-set. Or it could simply be down to how many activity instances are available per user, neither of which is reported in the paper. Be that as it may, these results again underline that the literature neither universally supports nor rejects our hypothesis that PSMs outperform PIMs on subject-dependent HAR performance.

2.1.4 Multi-Class Decomposition Methods for Human Activity Recognition

Many classification algorithms were originally designed to solve binary classification problems. To apply these binary algorithms to multi-class classification problems, the problem must first be decomposed into a set of binary classification problems. Then, a separate instance of the learning algorithm is trained for each of these binary problems. When a new sample is presented

to the system, it is passed to each of the trained classifiers and their outputs, which may be probabilities, are combined [Gal+11].

There are several methods for decomposing a multi-class classification problem into a set of binary classification problems [Gal+11; Par12]. The most popular of these are undoubtedly one-versus-all (OVA) and, to a lesser extent, one-versus-one (OVO). Another approach is based on error-correcting output codes (ECOCs), which may be constructed randomly, or learned from labelled or unlabelled data. Finally, there are hierarchical methods in which the classes are arranged in a tree or (in rare cases) even in a directed acyclic graph, which may be constructed randomly, learned from data, or constructed from common sense or domain knowledge. Such a hierarchical approach, which is often referred to as a *top-down* approach, is particularly appealing in application areas where the concepts (or classes) of interest are naturally arranged in a hierarchy, such as in HAR applications. There are examples of more or less formal class hierarchies in many other application domains—such as gene and protein function ontologies, music (and other artistic) genres, and library classification systems—and this has inspired researchers to develop hierarchical classifiers that excel at text categorisation, protein function prediction, music genre classification, and emotional speech and phoneme classification.

Each of these methods, which are described in more detail in section 5.1, represents the target concepts in a different way to the learning algorithm, which may or may not be beneficial in terms of predictive performance. In this subsection, we discuss the literature that investigates how they affect the predictive performance of classification algorithms. We structure our discussion in two parts. In the first, we focus on the more popular flat multi-class decomposition methods, such as OVA and OVO, and on multi-class decomposition methods that are based on error-correcting output codes. In the second, we discuss hierarchical multi-class decomposition methods, such as nested dichotomies and ensembles of nested dichotomies.

Joseph, Robbins, Zhang, and Rekaya [Jos+10], who combined OVO and OVA with a latent variable model, and compared the performance on two tumour classification problems from micro-array DNA data [Yea+01; Pom+02],

found that while OVO quite clearly performed better than OVA on one problem (by over 10 percentage points on average), OVA tended to perform better on the other, albeit only marginally. In 2011, Galar, Fernández, Barrenechea, Bustince, and Herrera [Gal+11] presented an empirical comparison of OVO and OVA, in which they combined OVO and OVA with SVM, decision trees, kNN, Ripper [Coh95], and a positive definite fuzzy classifier [CW03], and evaluated their performance on 19 publicly available multi-class data-sets. They found that OVO outperformed OVA in almost all cases, although rarely by more than one standard error. Raziff, Sulaiman, Mustapha, and Perumal [Raz+17] compared OVO, OVA, and ECOC (with random code matrices of varying size) in combination with decision trees to identify ($k = 30$) people from accelerometer data acquired via a handheld mobile phone, and found that OVO, which achieved 88% accuracy, performed better than either OVA or ECOC, which achieved 70% and 86%, respectively. They also found that when the width of the ECOC matrix was increased from k to $2k$, the accuracy increased by 11%. However, when the width was increased beyond that—to $3k$, $4k$, and finally $5k$—the rate of improvement slowed down to 2% to 3%. These studies show that while OVO is likely to perform better than OVA in most cases, it is not guaranteed to do so for any particular problem.

Hierarchical models in the form of nested dichotomies (a binary hierarchy or tree of binary classifiers) have long been a popular statistical tool for analysing polychotomous response variables [Fox97], where they are usually combined with the binomial logistic regression model to draw inferences about the relationships between predictors and the response. The link between the statistical theory of nested dichotomies (namely that the constituent nested dichotomies are independent) and hierarchical classification in a machine learning context was established in 2004, when Frank and Kramer [FK04] introduced the ensemble of nested dichotomies (END) and compared its performance to OVO, ECOC, and OVA on 21 publicly available data-sets. Besides confirming that OVO tends to perform better than OVA, they also found that ensembles of nested dichotomies were comparable to ECOC and more accurate than OVO when combined with decision trees, and comparable to OVO and more accurate than ECOC when combined with Logistic Re-

gression. Zimek, Buchwald, Frank, and Kramer [Zim+10] compared different hierarchical multi-class decomposition methods on four protein classification problems. Specifically, they assessed the performance of expert hierarchies built from machine-readable ontologies, ENDs, ENDs constrained by said ontologies, and non-binary expert hierarchies with an END at internal nodes (HEND). They found that while expert hierarchies improved the performance on simulated data, the HEND performed better on the data-set of real protein expressions. This shows that hierarchical multi-class decomposition methods that are based on domain knowledge can achieve better performance than randomly constructed hierarchies.

Due to the ease of constructing an intuitive hierarchy of increasingly detailed human activities, there are several papers that consider hierarchical classification for multi-class HAR. Mathie, Celler, Lovell, and Coster [Mat+04] and Karantonis, Narayanan, Mathie, Lovell, and Celler [Kar+06] develop hierarchical classifiers for multi-class HAR problems. Both papers independently develop each of the binary classifiers that constitute the hierarchical classifier, which makes them good examples of how a hierarchical approach can be iteratively refined. However, because neither paper considers alternative approaches, they do not help in answering the question how such hierarchical approaches compare to standard multi-class decomposition methods. Another paper that proposes a hierarchical HAR method is the work by Fortino and Gravina [FG15]. They propose a two-level hierarchy for fall detection consisting of a top-level classifer that detects falls which is followed by a second classifier to determine the severity of the detected falls. The top-level classifier runs on the IMU and applies a simple threshold to each accelerometer sample's cross-axial energy. Samples whose energy exceed the threshold signal a potential fall and trigger a second classifier to determine the fall's severity. The second classifier (kNN) is repeatedly applied over a period of 30 s to determine whether the user is in a lying or upright posture. Depending on whether or not, and how quickly, the user is able to regain an upright posture, the period is labelled as a green, yellow, or red severity fall. This is a good example of another advantage of hierarchical methods, namely their modularity, which makes it easy to design systems in which

different modules (classifiers) are executed on different devices. Unfortunately, this paper does not consider alternative (non-hierarchical) approaches either, and thus does not answer the question whether hierarchical multi-class HAR methods are competitive to other multi-class decomposition methods. Cho and Yoon [CY18] propose a two-stage approach to multi-class HAR which first classifies instances into "dynamic" and "static" activities, and then applies a multi-class convolutional neural network (CNN) to further discriminate among dynamic and static activities, respectively. Although the hierarchies from this paper have only two levels, the authors mention that using CNNs at internal nodes already significantly increased model complexity. They suggest that this could be remedied by replacing the top-level CNN with a simpler method. While they do not include other methods in their evaluation, they do assess their methods on two public benchmark data-sets. Their results are slightly (0.1 and 1.2 percentage points) better than the ones reported for the LSTM due to Ordóñez and Roggen [OR16], and for the DCNN due to Jiang and Yin [JY15], providing some evidence that supports the hypothesis that hierarchical methods can compete with non-hierarchical methods for multi-class HAR. Interestingly, their CNN erroneously labelled 655 instances of "walking" from one data-set as "standing". These errors have to be attributed to the top-level classifier alone because walking is a "dynamic" and standing a "static" activity, and because only one of the two second-stage classifiers—the one associated with the predicted top-level activity—is applied. All of these methods differ from nested dichotomies in that they predict a discrete activity at internal nodes via hard thresholding, only apply the classifiers that correspond to the predicted activity, and output a single predicted activity label in the end. A nested dichotomy, on the other hand, multiplies the probabilities of the internal nodes on the path to each leaf to predict a probability for each activity, rather than a single activity label. Nested dichotomies thus take not only the activity that classifiers predict as the most likely into account, but also the confidence that classifiers assign to each activity.

The non-probabilistic approach—trace the path of discrete "yes" or "no" predictions down the tree until hitting a leaf and return its class as the predicted label—appears to be the norm in the hierarchical classification

literature. None of the 74 papers—38 on text categorisation, 25 on protein function prediction, six on music genre classification, three on image classification, and one each on phoneme and emotional speech classification—reviewed by Silla and Freitas [SF11] in their 2011 survey of hierarchical classification used probabilistic hierarchies, opting instead for non-probabilistic hierarchies that discard their constituent classifiers' confidence in their predictions. Nevertheless, Silla and Freitas [SF11] found that hierarchical classification is a better approach to hierarchical classification problems than flat approaches, including not only OVO and OVA, but also inherent multi-class algorithms. More recently, in 2018, Silva-Palacios, Ferri, and Ramírez-Quintana [SFR18], experimenting with learned, rather than pre-defined, hierarchies across 15 multi-class benchmark data-sets (none of them HAR data) from the UCI machine learning repository [DG17], reported that probabilistic nested dichotomies clearly tend to outperform their non-probabilistic counterparts, albeit only by a small margin.

Unfortunately, none of the comparative studies of multi-class decomposition methods in the literature includes a HAR problem in their evaluation, and, because there appears to be no multi-class decomposition method that is dominant across all multi-class classification problems, we cannot assume that OVO, which tends to perform best in most domains, is going to also do so in the HAR domain. Furthermore, it can be argued that the concepts (activities), which HAR algorithms are trained to recognise, have a much stronger hierarchical structure than those targeted by most multi-class classification benchmarks, which may affect multi-class decomposition method performance. Moreover, none of the papers that do address the multi-class decomposition problem in a HAR context compares the performance of the proposed method to that of other multi-class decomposition methods such as OVA or OVO. Given the intuitiveness and popularity of hierarchical multi-class decomposition methods for HAR, and their inherent modularity and flexibility, it is important to study whether or not there is a trade-off between using a hierarchical multi-class decomposition method such as an expert hierarchy and using domain-agnostic multi-class decomposition methods such as OVO and OVA, and, if this is the case, estimate how much we stand to

gain (or lose) from using a hierarchical multi-class decomposition method that encodes HAR domain knowledge.

2.2 Device-Free Human Presence Detection from Wi-Fi Signal Data

In this section, we review the literature on device-free human presence detection from Wi-Fi signal data. We summarise the state-of-the-art for detecting mobile (e.g., walking) or stationary (e.g., sitting or standing) human presence from signals, such as received signal strength (RSS) or channel state information (CSI), that can be acquired from ordinary Wi-Fi networks. By "ordinary Wi-Fi networks" we mean those centred around an access point (AP) operating in infrastructure mode, which have a star topology at whose centre sits a single AP with which all clients associate and through which all transmissions must pass. We further limit our review to methods and results that have been evaluated in more than one room, or at least with multiple links. This section is organised as follows. The two subsequent paragraphs briefly outline the two types of Wi-Fi signals that are the raw materials for the device-free sensing techniques, whose discussion takes up the remainder and bulk of this section. This is divided into two parts. In the first (subsection 2.2.1), we examine methods that are based on RSS data and in the second (subsection 2.2.2) those based on CSI data.

In recent years, based on the fact that the human body absorbs and refracts radio frequency (RF) signals, researchers have explored device-free sensing with RF signals for a multitude of human sensing applications. Due to the ubiquity of Wi-Fi (IEEE 802.11) networks, much of this work uses Wi-Fi or other RF signals in the same frequency range (2.4 GHz to 5 GHz). While earlier device-free sensing papers [YMA07; MY09] relied on the total RSS, a fundamental signal in all RF communication, more recent papers increasingly abandon total RSS in favour of CSI, which is only exposed by those few Wi-Fi chipsets for which an experimental CSI-enabling driver exists. We are aware of only two families of chipsets for which such a driver has been released to

the public. Both of these drivers come in the form of patches to the Linux kernel. The older and much more popular of these drivers is available under the name "Intel CSI Tools" [Hal+11], and enables user-space access to CSI and ancillary data for Intel 5300 chipsets by means of a modified firmware and a software patch to the kernel's `iwl5300` module. The Intel 5300 had been the only chipset with a CSI-enabling driver from 2011 until 2015, when Xie, Li, and Li [XLL15] released the "Atheros CSI Tools." The Atheros CSI Tools do not rely on custom firmware and should work with any chipset that uses the kernel's `ath9k` module, which the Atheros CSI Tools patch when being installed. It should, therefore, support all types of Atheros 802.11n Wi-Fi chipsets that use that module. With a four-year gap between the first and second set of CSI tools, the "Intel CSI Tools" have, perhaps unsurprisingly, widely become known as simply the "CSI Tools," a convention we, too, adopt throughout this thesis.

CSI, which is part of the PHY layer since the 802.11n release of the Wi-Fi standard, captures how RF signals propagate along multiple paths from transmitter to receiver in the form of a time series of complex-valued 3D tensors of dimension $N_S \times N_{RX} \times N_{TX}$ where N_S, N_{RX}, and N_{TX} denote, respectively, the number of subcarriers, and the number of receiving and transmitting antennae. As such, CSI contains much richer information about the physical environment than total RSS, but at the cost of placing a much higher computational burden on algorithms operating on CSI data. In the IEEE 802.11 suite of standards which govern Wi-Fi networks, the total RSS is codified as the received signal strength indicator (RSSI). As its name suggests, the RSSI is an *indicator* of the total RSS across all of the receiver's antennae. As such, RSSI measures the relative RSS in arbitrary units, whose relationship to the RSS (which is customarily measured in decibel) differs from one Wi-Fi chipset to another. For a more detailed discussion of the differences and similarities between RSSI/total RSS and CSI, we refer interested readers to Yang, Zhou, and Liu [YZL13]. Other researchers have moved away from commodity Wi-Fi signals, and focused on using either a single RF link or node, [AK13; Sig+13; Sig+14; ZAK16; Zha+18] or an entire network of (e.g., RFID or ZigBee) RF nodes [WPH06; Yan+10; Xu+13; Pat+14; Rua+14;

YJ16; Zha+16; Kia+17], explicitly designed and deployed for the purpose of device-free sensing.

2.2.1 Methods Based on Received Signal Strength

Kosba, Saeed, and Youssef [KSY12] propose RASID, a device-free motion detection system that uses the RSSI of a number of "monitoring points," which are off-the-shelf Wi-Fi transceivers that actively scan the RF environment once a second to measure the RSS of any reachable APs. RASID takes an anomaly detection approach to human motion detection. As such, it does not require any examples of actual human motion, but is instead trained with a small amount of data collected when the target environment is devoid of humans. RASID's performance is evaluated in two testbeds. Testbed one consists of a bigger and eight smaller rooms of varying dimensions, all of which are connected by two narrow corridors and cover a total area of 16 m × 12 m. Testbed two encompasses two floors. The lower floor has an area of 12.6 m × 11.3 m and consists of four smaller rooms, and one large room which contains two APs and two monitoring points. The upper floor has a slightly smaller area of 12.3 m × 11 m and consists of three smaller and four medium-sized rooms, one of which contains the remaining monitoring point. All bar the smallest room are connected to a larger central atrium that houses the testbed's other two APs. In each of the two testbeds, the authors gather a total of 1.25 h which includes two traversals of testbed two and three traversals of testbed one. During each traversal, a single human walks at a normal pace along a predefined path that takes them through and around each of the testbed's rooms. RASID is trained/calibrated with data from the initial two minutes of the empty testbed, and tested on the remainder. The authors report F1-scores of 95.7% and 93.1% in testbeds one and two, respectively.

Zhou, Li, Xie, and Nie [Zho+19] use RSSI data from five Wi-Fi APs, whose transmissions are sampled once a second by three strategically placed Wi-Fi monitors, to detect whether human movement occurred and determine in which of the four detection regions the movement occurred. The detection

regions correspond to parts of a corridor leading up to a lobby, and the lobby room itself, and span an area of 49.3 m × 17.8 m. To reduce the interference of time-variant environmental noise in the RSSI, they minimise the maximum mean discrepancy (MMD) between the marginal distributions of the (labelled) training and the (unlabelled) test data, both of which were acquired in the same facilities but at different times. Then, both sets of data are transformed into the same subspace via the optimal transfer matrix, which is constructed from the minimum MMD. Fitting a learning algorithm (kNN, random forests, SVM) to the transformed data, the authors were able to detect human movement and the region in which it occurred with >97% accuracy. This is an improvement of, depending on the learning algorithm, between 7 and 22 percentage points compared to the accuracy achieved with raw data. The accuracy among algorithms with the transformed data differs by less than one percentage point, but the false alarm rate ranges from 0% with kNN, to 1.11% with random forests, to 4.21% with SVM. Unfortunately, the authors do not supply details about how the data they used to train and evaluate models in their experiments were selected.

2.2.2 Methods Based on Channel State Information

In a survey of CSI-based device-free human sensing published in 2019, Ma, Zhou, and Wang [MZW19] cite papers that use CSI data for presence and movement detection, localisation, heart and respiration rate estimation, and humidity estimation, with the majority targeting localisation, or activity or gesture recognition. Of the 157 papers, fifteen propose and evaluate methods for human presence detection, with predictive performance—mostly reported as accuracy, precision, or true (and false) positive rates—ranging from 85% to 100% with a mean of 94% and a standard deviation of 4.1 percentage points. Although many of them are discussed in this subsection, we shall limit ourselves to the most pertinent papers. In particular, we only cover methods that were experimentally evaluated with multiple links and testbeds, achieve excellent performance, or whose design confers unique added benefits such as significantly reducing the required calibration efforts.

Xiao, Wu, Yi, Wang, and Ni [Xia+12] propose FIMD, a motion detection scheme which relies on CSI data captured by one or more Wi-Fi clients. FIMD uses the two largest eigenvalues of the CSI amplitude's auto-correlation matrix of receivers' first antenna as input features. FIMD, like RASID, takes an anomaly detection approach. What sets FIMD apart is that it uses an unsupervised algorithm which does not rely on any *labelled* data. Instead, FIMD applies the density-based spatial clustering algorithm DBSCAN [Est+96] to a batch of CSI data to detect bursts. When FIMD detects a burst, the bursty instance is passed to a false alarm filter which examines whether or not its "feature value is isolated" from the two adjacent instances' and discards it as a false alarm if this is the case.

They separately evaluate FIMD in two testbeds, a 7 m × 11 m research lab and a 32.5 m × 1.5 m corridor, each of which is instrumented with an off-the-shelf Wi-Fi AP and a Wi-Fi client equipped with a CSI-capable three-antenna network interface card (NIC). In each testbed, they collected two hours of data. Human motion is represented by a single person walking randomly around the entire area of interest. Unfortunately, the authors do not report how their data are split between the empty and motion scenarios, which makes it somewhat difficult to interpret the results. The results are presented graphically in the form of receiver operator characteristic (ROC) curves. The authors report true and false positive rates ranging from >70% and ≤1% to 90% and >14%, respectively, in the lab. In the corridor, they report true and false positive rates of >90% and approximately 9%, respectively. In a separate evaluation they replace the feature used by RASID, which is based on the RSSI's standard deviation, with the maximum eigenvalue of the CSI auto-correlation matrix used by FIMD. This evaluation shows that RASID performs slightly better with the CSI feature than with the original RSSI feature. Interestingly, RASID—both with the CSI and RSSI feature—achieves a true positive rate that is comparable to FIMD's in the lab, but clearly (approximately 2 to 5 percentage points) better in the corridor. This is most easily explained by the fact that RASID, unlike FIMD, uses supervised learning insofar as it is calibrated/trained with data that are known to correspond to the empty area of interest. Unfortunately, Xiao, Wu,

Yi, Wang, and Ni do not report how these training data were selected, or in what proportion they stand to the data used to evaluate its performance.

Zhou, Yang, Wu, Shangguan, and Liu [Zho+13] propose a method to detect the presence of a mobile or stationary human in a clearly defined radius around a CSI-capable Wi-Fi receiver (client) with three antennae. Their approach, which they further refined in an extended journal paper [Zho+14], relies on a fingerprinting database against which new CSI samples are compared. Samples are compared via the earth mover's distance between their respective histograms over a sliding window. The fingerprinting database consists of one minute of data for each of the nine considered test conditions. The first of these is the normal condition in which no one is near the receiver. The remaining eight test conditions are when a stationary person is at one of eight different locations. Four of these locations are separated by 90° angles on a circle with radius 0.5 m centered on the receiver, the other four are located in the same way along a radius of 1 m. In addition to this fingerprint-based detection they also propose a threshold-based method that only requires data that correspond to the normal condition.

They evaluate their schemes separately for each link in the two testbeds, a relatively empty conference hall and a small cluttered computer lab, both of unspecified dimensions. A total of seven links are installed, four in the conference hall and three in the lab. Test data are sampled at 20 Hz for 30 s for each of the seven links, nine conditions, and nine individuals who partook in the experiment, adding up to a total of just under five hours. Half of the test data are collected with three additional people walking about the testbed room, but staying at least 2.5 m away from the link between the receiver (client) and transmitter (AP). Their fingerprint-based detection scheme achieves average false positive and negative rates of 7.9% and 4.8%, respectively, when an intruder is 0.5 m away from the receiver, and 6.9% and 6.4% when they are 1 m away. This corresponds to true positive rates of 95.2% and 93.6% for the 0.5 m and 1 m radius, respectively. The link-wise false positive rates for both radii remain near zero for all but one link, where they rise to about 12%. This shows that while most links rarely raise false alarms, others can have false alarm rates of 10% or more. The corresponding false negative rates

approximately range from 3% to 12%, corresponding to true positive rates ranging from 88% to 97%. The threshold-based scheme achieves average false positive and negative rates of 6.8% and 7.9%, respectively, corresponding to an average true positive rate of 92.1%. Here, the approximate link-wise false negative and positive rates range from 3% to 15% and from 5% to 12%, respectively, corresponding to true positive rates between 85% and 97%.

Qian, Wu, Yang, Liu, and Zhou [Qia+14] propose PADS, a CSI-based motion detection system, which they further refine and explore in an extended journal version [Qia+18] of their 2014 conference paper. PADS uses data from a wireless link between a Wi-Fi AP and client, both of which are equipped with a three-antennae CSI-enabled wireless NIC. In contrast to most other CSI-based human sensing methods which only use the CSI amplitude, PADS uses both the (real) amplitude and (imaginary) phase components of the CSI. Specifically, PADS extracts the three largest eigenvalues of the pre-processed phase's and amplitude's auto-correlation matrices. The six eigenvalues (three from the amplitude and three from the phase) are then used as the input features for a SVM classifier. The authors argue against an unsupervised clustering approach, such as the one employed in FIMD, and in favour of a supervised learning approach by noting that although such a clustering approach does circumvent the need for *labelled* training data, it still requires a substantial data-set to form accurate clusters and "assumes that at least two states are involved in each group of measurements to be processed." They evaluate two different flavours of PADS and FIMD. The first PADS variant is the original version of PADS [Qia+14], PADS-LT, which employs linear transformation to pre-process CSI phase data. The second variant, termed PADS-PD, calculates and unwraps the phase difference between antenna pairs to remove random noise. Both PADS-LT and PADS-PD pre-process CSI amplitude by identifying and removing outliers with a Hampel filter.

They acquire their evaluation data at different times from four testbeds, a $30\,\mathrm{m}^2$ meeting room, a $2\,\mathrm{m} \times 27\,\mathrm{m}$ corridor, and a classroom and two labs with approximate areas of $80\,\mathrm{m}^2$. Depending on the testbed, the AP is placed at a height varying from 1.2 m to 2 m with the client located 2 m to 7 m away from it. Although the authors do report that both line-of-sight (LoS) and

non-LoS (NLoS) links were included, they do not supply any further details on the topic. In total, they collect over an hour of data during which a single human is walking through each testbed at approximate velocities of 0.5 m/s, 1 m/s, and 2 m/s along a pre-defined path which uniformly traverses the area of interest. This hour constitutes the positive examples—instances of a moving human—in their data-set. They acquire the same amount of data during which either no one, or only stationary people, are present in the area of interest, providing the negative examples—instances without moving humans. They train a single SVM classifier with part of the data from all testbeds, and separately make predictions with the remaining data from each testbed. The experiment is repeated multiple times while varying the sampling frequency from 50 Hz to 1000 Hz, the sliding window size from 0.2 s to 2 s, and the number of features (eigenvalues) from 1 to 10.

In these experiments PADS-PD, which consistently outperforms both PADS-LT and FIMD, achieves average (across testbeds) true and false positive rates of 99% and 0%, respectively, when sampling at 200 Hz and using six eigenvalue features extracted along a sliding window of 2 s. Their results show that larger window sizes correspond to better performance for all three methods, although not to the same degree. At 1 s the rate of improvement for PADS-PD visibly flattens, and at 2 s a similar flattening can be observed for the other two methods' curves. Even with the smallest window size of 0.2 s, PADS-PD still achieves a true positive rate of 94%. PADS-LT on the other hand, which does nearly as well as PADS-PD when using two-second windows, achieves a mere 86% in this case. And FIMD, which trails the others by 2 to 3 percentage points with 2 s windows, achieves only 83%. The impact of the sampling/transmission rate follows a similar pattern, albeit of a smaller magnitude, for PADS-LT and FIMD, but is indiscernible for PADS-PD. For PADS-LT, the true positive rates range from just below 96% with a transmission rate of 50 Hz to 98% with 1000 Hz, whereas they range from 90% with 50 Hz to 94% with 1000 Hz for FIMD. The number of features, it turns out, has a much bigger effect on the true positive rate than either the window size or transmission rate for all three methods. FIMD achieves 73% with one, 95% with three, 97% with five, and 99% with ten features.

PADS-LT achieves 88% with one, 98% with three, and 99% with four features, after which improvements are barely discernible. PADS-PD achieves 82% with one, 98% with two, and 99% with three features, after which the curve flattens to such a degree as to be indistinguishable from a straight line. They cross-validate PADS-PD with the three different walking velocities, training a separate model for each velocity and evaluating it on the two other velocities. The results show that, regardless of which walking velocity PADS-PD is trained with, it is more sensitive to faster than slower walks. They also show that PADS-PD is more sensitive, across all three velocities, if it is trained with the slower walking speeds. Hence, the worst performance ($\approx 93\%$) is achieved when PADS-PD is trained with people walking at a fast pace (2 m/s) and tested with people who are walking at a slow pace (0.5 m/s), and the best ($\approx 99\%$) when it is trained with people walking at a slow pace and tested with people walking at a fast pace.

Zhou, Yang, Wu, Liu, and Ni [Zho+15] propose an intricate scheme that uses a sample of CSI calibration data collected without anyone near the link to fine-tune a Wi-Fi link's sensitivity to human presence at run-time. It works by estimating the sensitivity of different RF propagation paths and CSI subcarriers and use it to weight CSI amplitude data. The weighted data are then used to calculate the (euclidean) distance between new CSI samples and the calibration data. If the distance exceeds a certain threshold—which is empirically determined by analysing the ROC curves for an evaluation data-set of both positive and negative examples—it is classified as human presence.

They evaluate their scheme in two indoor tesbeds, an 8 m × 6 m classroom equipped with three Wi-Fi links of varying lengths, and a 4 m × 3 m two-person office equipped with two links. Each link consists of an off-the-shelf Wi-Fi AP with a single antenna, and a CSI-enabled Wi-Fi client with three antennae. For each link, human presence is evaluated at nine locations, arranged in a 3 × 3 grid that equally divides the distance between transmitter and receiver, with three locations on and three more to either side of the link. During the data collection, up to five people are permitted to work at their desk and walk around the room, staying at least 5 m away from the links.

Temporal diversity is taken into account by "pausing for 5 minutes before measuring the next 5000 packets and repeating the measurements both in the daytime and at night, and after two weeks." They report an average true positive rate of 92%, ranging from 90% to 100% across the five different links, and an average false positive rate of 4.5%.

Wu et al. [Wu+15] propose DeMan, a method to detect both moving and stationary humans from a single Wi-Fi link. DeMan consists of two detection modules, one designed for detecting mobile (i.e., walking) and the other for detecting stationary humans, and a motion interference indicator to determine which detection module should be deployed for any given sample. The motion interference indicator entails comparing the variance of the sample (window) under consideration against the variance with and without human movement, both of which are estimated from training data. Borderline cases—those falling within a designated critical zone that is near both variance thresholds—are subjected to both mobile and stationary human presence detection, whereas instances that clearly lie on either the mobile or stationary side of the thresholds are only subjected to the corresponding detection module. DeMan uses the largest eigenvalue of the amplitude's and phase's auto-correlation matrix to detect mobile presence, i.e., the presence of a walking person, via a SVM classifier. To detect stationary presence (i.e., the presence of a sitting or standing person), DeMan applies a bandpass filter to retain only the frequency range (0.15 Hz to 0.7 Hz) that corresponds to that of normal human breathing, then fits a sinusoidal model to each subcarrier's filtered signal to estimate the frequency and amplitude of the wave's dominant component. After identifying and removing outliers via least median squares regression, the remaining estimates are averaged across subcarriers. If there is no breathing human in the monitoring region, then the estimated sinusoidal's amplitude peak tends to be near zero, but if a breathing human is present then the peak tends to be closer to 1 dB.

DeMan is evaluated with data from two testbeds, a 6 m × 8 m lecture room, and a 12 m × 6 m open office. In total, there are five Wi-Fi links, each consisting of an off-the-shelf Wi-Fi AP and client. Three of them are in the smaller classroom, and the remaining two in the larger office. One of

the latter two links is obstructed by a divider (of unspecified material and thickness) between two adjacent desks. The data consist of three categories. The first, in which a person walks along a pre-defined path that traverses the entire area of interest, corresponds to mobile presence. The second, where a person stands or sits for 2 min at each location in a uniform grid which spans the area of interest, corresponds to stationary presence. The third, in which the area of interest is vacant of people, corresponds to no presence. Each link is associated with its own area of interest which covers a rectangular region that is 2 m wide and as long as the link which it is centered on. In total, the data consist of about 8 h, evenly divided among the three categories (mobile, stationary, and no presence). The authors state that the SVM which is used to detect mobile presence is trained with "a portion of measurements," but do not specify which, or how much, data are used for this. Nor do they provide details about the data from which the motion interference indicator thresholds are estimated. Given that they "do not need to calibrate the parameters for each different scenario over different time" and that no specific links or testbeds are mentioned in this context, it seems most likely that the calibration/training data-set combines a small portion of data from each link. In these experiments, DeMan achieves true positive rates of 93.82% and 94.82% for stationary and mobile presence, respectively, and a true positive rate of 94.08% across both types. These results come with a false positive rate of 3.75%, a miss (i.e., false negative) rate of 6.67% and 5.18% for stationary and mobile presence, respectively, and a true negative rate of 96.25%.

Palipana, Agrawal, and Pesch [PAP16] use one-minute batches of CSI data, sampled at 1000 Hz from a single link between two three-antenna Wi-Fi nodes, to detect the presence of a stationary person at multiple locations in a research lab. Although their evaluation is limited to data from a single room, they propose a CSI amplitude feature which we have not yet encountered. They use the eigenvalues of the second and third principal components (PCs), which are obtained by applying kernel principal component analysis (PCA) to non-overlapping 20 s windows. The PC features have a major advantage over the eigenvalues of the (temporal) auto-correlation matrix (or matrices, if we consider both amplitude and phase), and that is that their computational

complexity tends to be lower. The most expensive operation for both the PC and auto-correlation features is finding the eigenvalues of a positive semi-definite matrix, which has time complexity $O(n^3)$. The main difference is that the dimensionality of that matrix is fixed by the number of subcarriers (the CSI Tools export 30) and antenna pairs (ranging from one to nine in a typical commodity Wi-Fi link) for the PC features, but depends on the sampling/transmission rate and window size for the auto-correlation features. The PCs eigenvalues are compared to the largest corresponding eigenvalue from a sample of CSI data acquired with the room empty. They achieve true and false positive rates of 85.3% and 2.5%, respectively, with linear PCA, 89.4% and 0% with Gaussian kernel PCA, and 62.6% and 2.5% with polynomial kernel PCA.

Li et al. [Li+17] propose AR-Alarm, a threshold-based method for detecting intrusions (i.e., mobile human presence vs. the empty room) designed to be robust to environmental changes. It relies on the standard deviation, calculated along a sliding window, of the phase difference between two antennae as the main feature. To make it more robust to different environments, the authors divide it by its historical maximum in a vacant environment. This normalising denominator is adaptively updated whenever a static environment is detected. To demonstrate their method's efficacy, they initialise its thresholds based on labelled data acquired in one office room, and evaluate it in a—presumably different—3 m × 4 m office and a 6 m × 6 m meeting room. The data acquired in these rooms include repeated human presenc in each cell of a 1.5 m × 1.5 m grid that covers the rooms' entire areas. They also move big pieces of furniture such as bookcases, sofas, and tables around the room, noting that these changes have little effect on the system's performance. In these experiments, AR-Alarm achieves true and false positive rates, respectively, of 93.3% and 1.6% in the smaller office, and 98.1% and 2.8% in the larger meeting room.

Similarly, Zhu, Xiao, Sun, Wang, and Yang [Zhu+17] propose R-TTWD, a CSI-based method that is designed to be robust to environmental changes without relying on data from the deployment environment. R-TTWD uses the mean first-order difference of the eigenvectors corresponding to the second,

third, and fourth PC as features as inputs to a SVM classifier. They argue—and corroborate with their experimental observations—that features based on the (temporal) auto-correlation, such as those used in FIMD [Xia+12], PADS [Qia+14; Qia+18], and DeMan [Wu+15], "do not perform well in trough-wall scenarios." To further mitigate environmental effects on the CSI signal, they first remove outliers via a Hampel filter, then further sanitise it with a wavelet-based noise filter, before performing PCA to find the PCs and corresponding eigenvectors. They assess R-TTWD by training the SVM with data from a 6 m × 9 m meeting room, and applying it to data from a 5 m × 9 m office. In each scenario, the data come from a single through-wall link between a Wi-Fi AP and a three-antenna laptop with the CSI Tools. In the meeting room, the same single-antenna AP is used for all experiments. In the office, parts of the data were acquired with a single-antenna AP while others were acquired with a two-antenna AP. In these experiments R-TTWD achieves average true positive and negative rates of 100% and 95.6%, respectively.

2.3 Conclusions

In this chapter, we have reviewed the literature on the topics pertinent for this thesis. We summarised the state-of-the-art in HAR from IMU data, and discussed papers that elucidate the relationship between subject-dependent and -independent HAR performance of micro and macro models, which we termed PSMs and PIMs, respectively. We reviewed works that build personalised HAR models with a limited amount of end-user data, and discussed the literature on how different ways to encode the concepts (e.g., activities of interest) for machine learning algorithms affect predictive performance. Finally, we reviewed the literature on device-free human movement detection from Wi-Fi signal data.

In our review of the HAR literature on the subject-dependent and -independent performance of micro and macro models, and on personalised HAR models (in subsections 2.1.2 and 2.1.3, respectively), we discussed papers pertinent to the question whether micro models (i.e., PSMs) are a better choice for known users than the corresponding macro model (PIM). Our findings

can be summarised as follows. Only three papers [WL12; BG17; Fer+20] report the subject-dependent performance of both PSMs and PIMs. One of them [WL12] reports that PSMs outperform the corresponding PIM by 2% to 30%, depending on which of the eight learning algorithm is being evaluated, on a single data-set. Another [BG17] reports that PSMs outperform the PIM by 1 to 25 points, depending on which of the four data-sets the learning algorithm is applied to. And Ferrari, Micucci, Mobilio, and Napoletano [Fer+20] found that PSMs, on average, perform 15 points *worse* than the PIM across three data-sets. Moreover, the main topic of two papers [BG17; Fer+20] is personalising HAR models with a small amount of user-supplied data, with PIMs and PSMs serving as baselines. As such, they are trained with only a small percentage of user data, and evaluated on the remainder, which provides little information about how the two compare when using all the available data. Therefore, the question remains open whether micro models (i.e., PSMs) really are a better approach to HAR for populations of known users than their macro counterpart, the much more prevalent PIM.

In our review of the literature on multi-class decomposition methods for HAR, we discussed papers that directly compare the more popular non-hierarchical multi-class decomposition methods, unfortunately not on HAR data-sets. The results in this literature shows a tendency for OVO to perform (slightly) better than OVA. Literature that compares hierarchical and other decomposition methods is much sparser. We discussed one paper that proposes and compares ENDs with other multi-class decomposition methods, finding that ENDs perform comparable or better than standard multi-class decomposition methods. Another paper directly compares domain-agnostic and domain-driven hierarchical approaches on four protein classification problems, reporting that hierarchies which incorporate domain knowledge perform comparable or better than domain-agnostic approaches. Surprisingly, we found non-probabilistic hierarchies to be far more popular than probabilistic ones, despite evidence that the latter type is superior. Finally, we looked at the literature on hierarchical classification for HAR. We saw demonstrations of some of the benefits of hierarchical approaches for HAR, namely iterative development and modular design, but found little evidence on how these

methods compare to standard domain-agnostic multi-class decomposition methods.

Our review of the literature on device-free human presence detection from Wi-Fi signal data shows that it is possible to learn to detect human movement from Wi-Fi signals, be they RSS or CSI, collected from one or more links in one building or room at one time, and use it to detect human movement from unlabelled Wi-Fi signals collected from the same links in the same facility, but at a different time. Moreover, it also demonstrates that such a learning scheme is not necessarily restricted to the facilities for which it was initially developed, but can be calibrated to detect human movement in different rooms or buildings. This has been demonstrated by many papers whose authors collect labelled Wi-Fi signals in multiple facilities and use part of each facility's (or all facilities') data to calibrate their learning scheme, while the other part is held out to evaluate its efficacy. However, only few papers assess how well their human presence detection schemes fare when deployed in facilities which they have no prior knowledge about nor control over, using whatever Wi-Fi signals are available there at the time. Finally, all but two [Wu+15; Qia+18] of these papers only consider micro (single-link) models, and none of them directly compare micro and macro models.

In the next chapter, we present a HAR method that recognises up to seventeen activities for monitoring emergency first responders. In doing so, we shall encounter some of the issues that motivate chapters 4 and 5, and develop the classifiers that are used in these subsequent chapters.

Chapter 3

Wearable Human Activity Recognition for Emergency First Responders

This chapter[1] is an initial application example of our work with wearable human activity recognition (HAR), which we present in chapters 4 and 5. In this chapter we also develop the methods that are used and extended in the two following chapters. This chapter brings the main issues to the fore that motivate much of the work we discuss in these latter chapters. Namely, the difficulty of choosing appropriate activities of interest, and the question whether the monolithic single macro-model approach that dominates in the HAR literature, and which we call a person-independent model (PIM) is the best way of training HAR models for a stable population of users who can be identified at prediction time.

We investigate Support Vector Machines (SVMs), k-Nearest Neighbours (kNN), and gradient boosted ensembles of decision trees (GBTs) for recognising up to 17 different human activities that are relevant for monitoring first responders during emergency response operations. The SAFESENS (Sensor

[1]The material in this chapter was published in two separate papers, titled "Human Activity Recognition for Emergency First Responders via Body-Worn Inertial Sensors" [Sch+17a] and "Sensor and Feature Selection for An Emergency First Responders Activity Recognition System" [Sch+17b].

Technologies for Enhanced Safety and Security of Buildings and its Occupants) project [TKO15] developed a novel location-tracking and monitoring system for firefighters and other first responders which makes that information available to them. The system monitors firefighters via wireless-enabled inertial measurement units (IMUs), attached to the straps of the self-contained breathing apparatus. Data are streamed from the IMU to a smartphone, carried by each firefighter, where an application buffers the data for 10 seconds before transmitting them in one batch to the command & control centre. In the command & control centre, the data are used to show the officers where their firefighters are and timely clues about what they might be doing. The system is designed to work reliably in the harsh and unpredictable conditions of emergency situations, and be resilient if pre-deployed infrastructure fails.

Much of the HAR literature is devoted to activities of daily living, especially for monitoring the elderly or other populations at risk, and there have been only a few attempts to extend HAR to the dynamic activities and environments typical of emergency first response operations. Frank, Diaz, Robertson, and Sánchez [Fra+14] develop a method for recognising safety relevant motion activities from inertial sensor data. Their thirteen activities of interest include not only typical activities of daily living such as walking, sitting, and standing, but also falling, lying down, and two types of crawling, which were chosen such that "professional users in safety relevant situations, like first responders or armed forces, can benefit from it." They propose a dynamic Bayesian network, whose structure is learned from data via the K2 algorithm [CH92], a process which kept the authors' computer(s) busy for fifteen days. The transition matrix which governs the dynamic part of the Bayesian network is developed according to "bio-mechanical expert knowledge," since a "first approach based on statistics of the data set did not show satisfactory results." Twenty-two features are extracted from the IMU signals along windows which are 0.25 s to 2.56 s long, depending on the feature. The network is learned and evaluated with a data-set that the authors collected for this purpose. It consists of 2 hours and 37 minutes, and covers 20 subjects, each of whom is equipped with a belt-mounted inertial measurement unit. Although overall accuracy is 82% and most activities' F1-scores range from 81.5% to 90.9%,

some are as low as 68.8% (Walking Upstairs) and one (Jumping) as high as 94.9%.

Ahmed, Frank, and Heirich [AFH15] extend the method proposed by Frank, Diaz, Robertson, and Sánchez [Fra+14], with the aim to handle multiple sensor positions. In particular, the inertial body frame is automatically updated whenever acceleration data indicate an upright dynamic activity, such as walking or running. To evaluate the approach, the authors acquire data from 19 participants following the protocols from Frank, Diaz, Robertson, and Sánchez [Fra+14], but with the sensor being placed once in each of five locations—viz. belt, pocket, hand, texting, and phoning—for each activity. The two data-sets are combined, and used to evaluate the approach via stratified (over activities and sensor placements) ten-fold cross-validation (CV). The results for some of the activities are rather poor. Recall and precision among dynamic activities range from 54.68% to 100% and from 55% to 81%, respectively, and precision ranges from 68% to 89% among static activities.

Unfortunately, Frank, Diaz, Robertson, and Sánchez [Fra+14] only evaluate their model with the same data it is trained with. Those results, therefore, say nothing about the model's ability to generalise to new data, even if they are obtained from the same group of users. The results presented by Ahmed, Frank, and Heirich [AFH15] are more informative because they were obtained via CV. This makes them a reasonable estimate of the expected predictive performance if the model is applied to new data from the same group of users. They do not, however, speak to the model's ability to generalise to new users. Nor is their performance anywhere near the results we saw in our review of the state-of-the-art in HAR with IMU data in subsection 2.1.1. Furthermore, the special equipment that is worn and carried by first responders such as helmets with face shields, breathing apparatuses, and heavy boots affects their every movement. This makes it unlikely that a HAR model pre-trained with data from an appropriate set of activities, performed by unencumbered users, would perform well if deployed in an emergency first response operation—if such a model and data-set existed. We show that our GBT is able to accurately and reliably distinguish among these 17 activities, using gyroscope and accelerome-

ter data from a single IMU, with subject-dependent and subject-independent mean absolute errors (MAEs) of less than 1% and 4%, respectively. Although HAR algorithms that use boosting exist [BS09; Les+05], the GBT algorithm in its canonical formulation [HTF09, ch. 10] had not previously been tuned and evaluated in this context.

3.1 Methods

We selected 17 activities in consultation with collaborating firefighters: two types of crawling (on hands & knees, and military style on one's stomach), duck walking, falling, two types of jumping (on and off a chair), three types of running and walking (horizontally, up/down the stairs) and five static postures: being on one's hands and knees (all 4s), standing, sitting, crouching, and lying down (e.g., after falling). Figure 3.1 shows examples of firefighters performing some of the activities. There was considerable discussion what activities of interest are the most appropriate. We knew that the more activities we included, the more complex the data acquisition protocols, experiments, and analysis were going to become, and the more the system's overall predictive performance and interpretability were going to suffer. In our work with the firefighters we opted to consider as many activities as was practical, and then group them in different ways to create three additional, simpler HAR problems. The first of these, the "move-type/lie" problem, consists of seven target classes (activities): All types of crawling, duck walking, falling, lying down, all types of running, all types of walking, and all the static postures. The next problem, the "move-type" problem, differs from the move-type/lie problem in that lying down is used as an additional static posture, reducing the total number of classes that are to be predicted to six. The fourth problem is the binary problem of discriminating between falls and everything else.

3.1.1 Experimental Design and Data Acquisition

We recruited eleven volunteers (all male, age: 20 to 34 years) via email and word of mouth from our institution's staff, each of whom met the firefighter

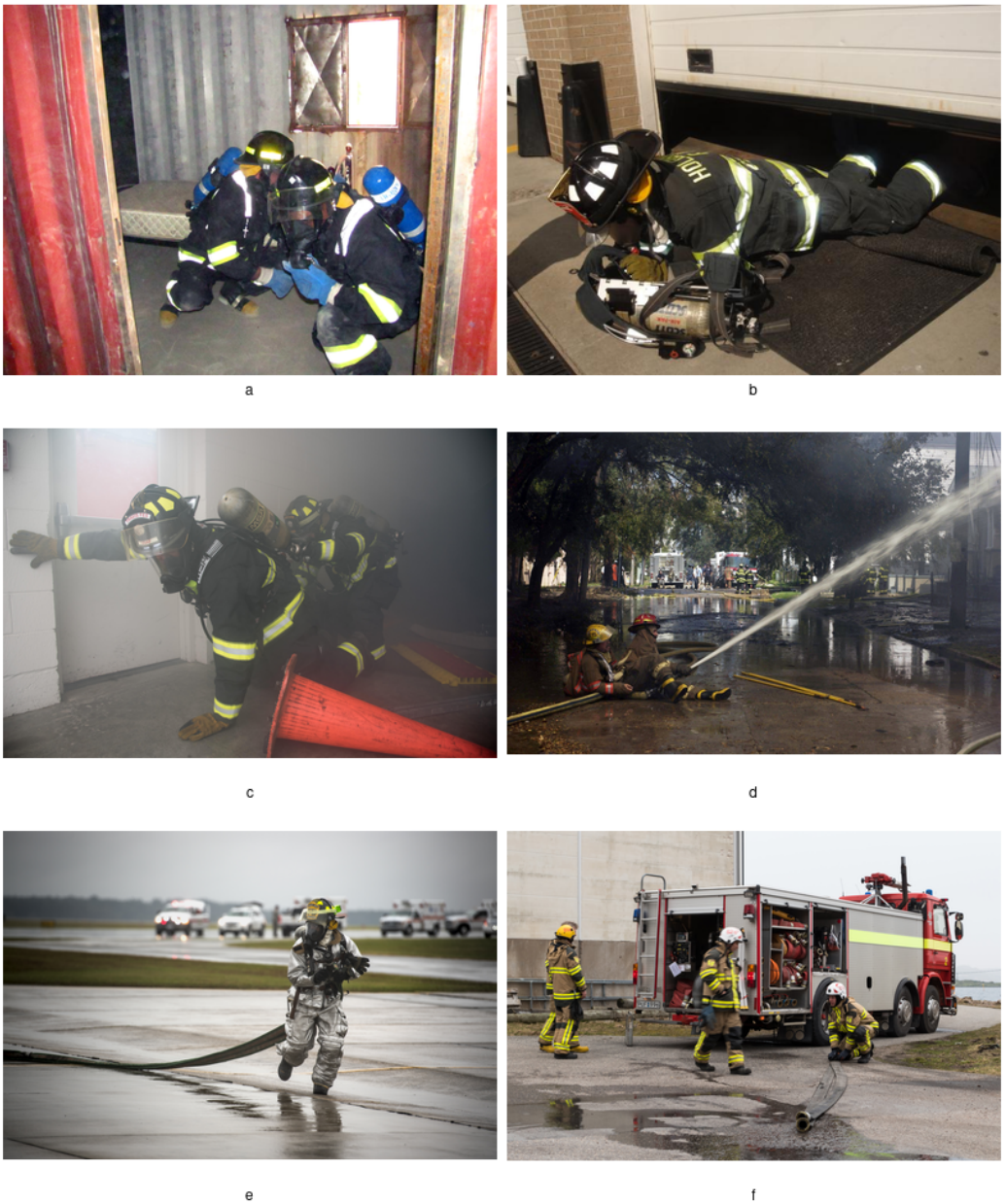


Figure 3.1: Firefighters engaging in a) duck walking, b) military crawling, c) crawling on hands & knees (front) and duck-walking (back), d) sitting, e) running, and f) standing, walking, and crouching. Source for panels a and c–e: U.S. Air Force, b: Holbrook Fire Department, and f: W. Carter.

eligibility criteria: aged between 18 and 37, at least 1.66 m tall, a body-mass index of 20–30, no problems with eyes, ears, or teeth, and of healthy and robust physical constitution. The experiments followed the ethics procedures in place at the Tyndall National Institute at the time of the experiment. Volunteers were briefed on the purpose and content of the experiment, as well as on data management and anonymity procedures, and signed a consent form. Participants were invited, one at a time, to our lab in the buildings of the Tyndall National Institute, where they were instructed to perform several supervised trials of each activity.

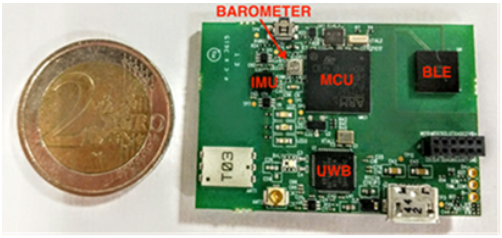


Figure 3.2: Inertial Measurement Unit

To simulate some of the constraints imposed by the firefighting gear we asked participants to wear heavy boots, and carry 13 litres of water in a backpack to simulate the weight of the self-contained breathing apparatus that firefighters carry during operations. One of the backpack's shoulder straps served to hold the IMU in place. The IMU, developed by the SAFESENS project, is equipped with a high-performance low-power 168 MHz 32-bit microprocessor with 1 Mb of flash memory and 192 Kb + 4 Kb of RAM, a bluetooth low energy (BLE) and an Ultra-Wideband (UWB) communication module, a rechargeable battery, sensors for barometric pressure, humidity and (internal and external) temperature, and a triaxial accelerometer, gyroscope and magnetometer. Inertial sensors are wired to the micro-controller through the I2C communication, while the environmental sensor adopts the SPI. The platform measures 44 mm × 30 mm × 8 mm without battery. Figure 3.2 shows a picture of the IMU board, which was housed in a 3D-printed case during the experiments. Sensor data can be transmitted wirelessly (via bluetooth), or logged to a removable Micro SD card. Our data were logged to the SD card at a sampling rate of 30 Hz. Other materials used were a chair for jumping on and off, a treadmill, and an inflatable mattress for falling and lying down. To aid with labelling the collected data, trials were timed by the experimenter, and participants

instructed to tap the IMU before and after each trial. To avoid potential bias in the data, the sequence in which the tasks were performed by each participant was randomised. For each trial, participants were further instructed to enact a (randomly chosen) variant of the task.

Falling

Participants were instructed to stand beside the mattress, then fall onto it, lie still for a moment, get up, and assume the starting position. For each trial, they were to fall either *forward*, or to the *side*.

Jumping

Participants were instructed to stand in front of (or on) the chair, jump onto (or off) it, pause for a moment, and finish by getting back in the starting position.

Horizontal Walking, Crawling, and Duck-Walking

Participants were instructed to move around the hallway and room in the specified manner for one minute per trial, or until they felt exhausted. For each walking trial they were to walk at either *slow*, *regular*, or *fast* speed.

Horizontal Running

Represented by two tasks: running on the treadmill (to capture running at steady velocity), and in the hallway (to capture turns and realistic accelerations). For treadmill running, participants were asked to run at 7 km/h, 10 km/h, or 12 km/h for 90 seconds. For hallway running, participants were instructed to run from one end of the hallway to the other at either *slow*, *regular* or *fast* speed. Each performed as many hallway running trials as needed to obtain 90 s of data.

Walking and Running, Up and Down the Stairs

Participants were instructed to position themselves at the top or bottom of the staircase, and then walk (or run) down (or up) the stairs at *slow*, *regular*, or *fast* speed, stop, and return to the starting position. Each participant performed as many trials as needed to obtain 90 s of data.

Static

The static tasks, or postures, are standing, sitting, crouching, all 4s, and lying down. For these tasks, participants were instructed to assume the position for one minute per trial. All static tasks, with the exception of crouching and all 4s, had designated variants. For standing, they were to either stand *upright*, *bent* forward, or *leaning* against the wall. For sitting, they were to either sit in normal position on a *chair*, upright on the *floor*, or on the floor with back or shoulder *leaning* against the wall. Finally, for lying, they were to lie either face-down on the *front*, or on the *side*.

3.1.2 Data Pre-Processing

The collected data are prepared as follows. First, the coordinate systems are aligned to conform to the same notion of up and down. Then we apply a median filter with a window size of 3 samples to smooth the signal, and resample the smoothed signal to its mean sampling frequency. If the original signal was not sampled with a constant frequency due to potential hardware limitations, then the resampled signal will contain gaps. These gaps are filled by linear interpolation, which is a reasonable approximation in the absence of further information.

Next, we replace each of the accelerometer's channels (x, y, z) with two derived features: its gravity and body component. The accelerometer captures acceleration from two sources: the earth's gravitation, and the movement of the IMU and its wearer. Because it is those movements we are interested in, we separate the two components with a low-pass filter as described in [Kar+06]. Afterwards, the original accelerometer signal contains no additional

information and is not used further. Finally, the signals are segmented into 3 s sliding windows with 1 s overlap. A duration of 3 s was chosen because that is long enough to capture even the slowest step, crawl, or fall, yet short enough to provide timely clues to the firefighters. The resulting data-set consists of 16 621 windows (instances), distributed as follows: all 4s: 5.8%, crouch: 4.4%, sit: 9.8%, stand: 8.6%, lie: 6%, crawl (hands & knees): 6.1%, crawl (mil.): 5%, duck walk: 4%, fall: 0.6%, jump off/on: 0.9% each, run hallway: 2.9%, run treadmill: 9.4%, run up: 5.6%, run down: 5.7%, walk horizontally: 5.9%, walk down: 8.2%, and walk up: 10.2%.

3.1.3 Feature Extraction

We extracted seven time-domain features—mean, sample standard deviation, skew, kurtosis, inter-quartile range, signal magnitude area, and pairwise correlations between each sensor's x, y, and z channels—and two frequency-domain features—spectral power entropy and peak-power frequency (PPF)—which have proven useful in previous HAR applications, from the gyroscope, and the gravity and body acceleration signals. Most of the features are statistical (e.g., mean, skew) and follow their usual definitions. The signal magnitude area, which has proven useful for detecting periods of physical activity in previous HAR work [Kar+06], was extracted from the gyroscope, and from the body and gravity acceleration signals. Both the spectral power entropy and PPF are popular frequency-domain features; both rely on a uniform sampling rate, and an estimate of the power spectral density. For the PPF the power spectral density was estimated via Welch's method and for the spectral power entropy via the periodogram. The spectral power entropy was then calculated following Ermes, Pärkkä, Mäntyjärvi, and Korhonen [Erm+08]. Figure 3.3 depicts examples of three activities' IMU signals and features. The first column (a) shows the raw accelerometer and gyroscope signals, and an assortment of features extracted from them, with the subject walking up the stairs. The second column (b) depicts the same signals and features, but with the subject walking down the stairs, and the third column (c) shows them with the subject running on the treadmill.

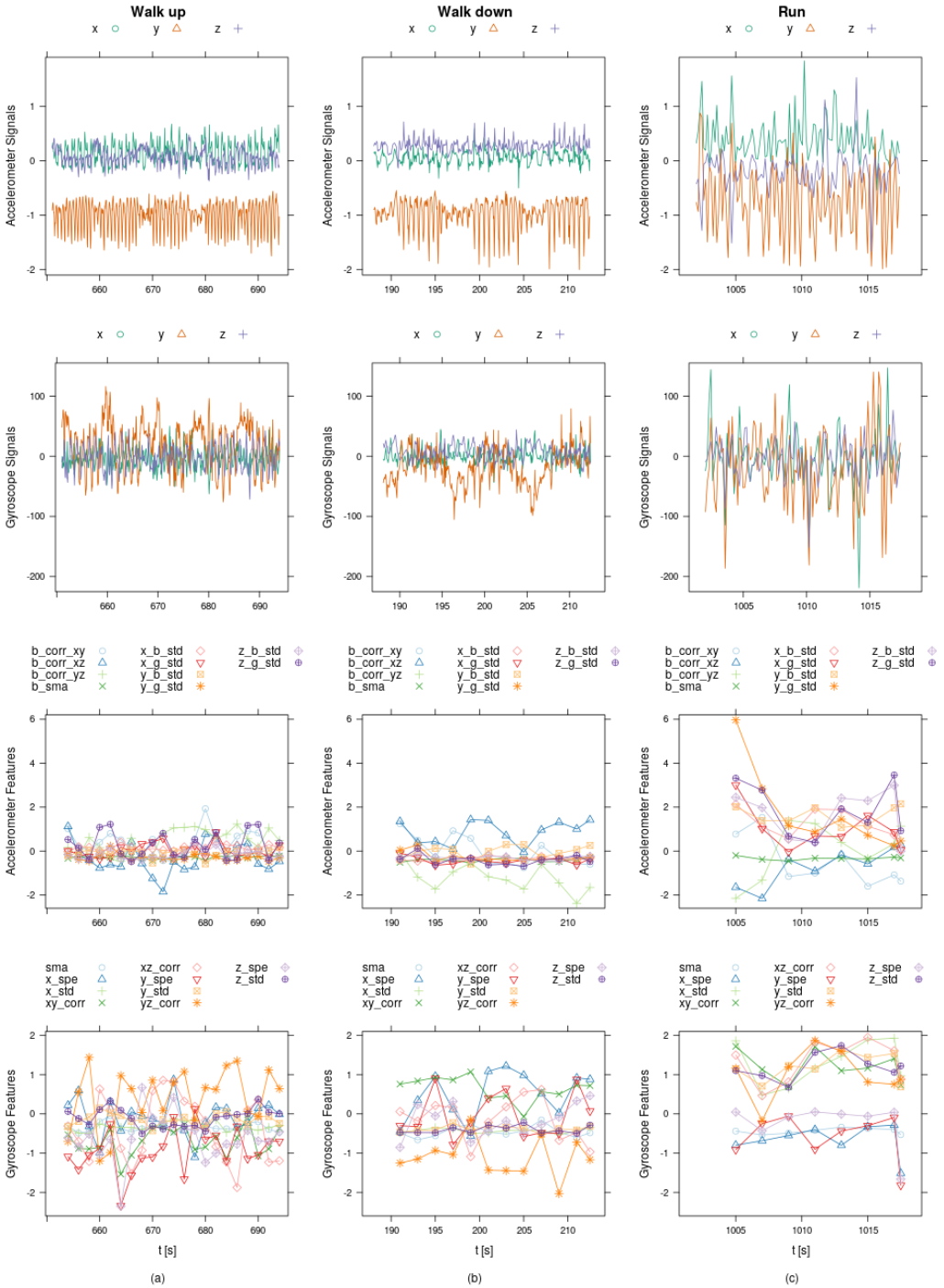


Figure 3.3: Inertial signals (two top rows) and assorted (standardised) features (two bottom rows) for three examples of walking up (a) and down (b) the stairs, and running on the treadmill (c)

3.1.4 Algorithm Tuning

We used the same procedure to separately tune each algorithm for each of the four problems. The procedure has been designed, following current best practices from the HAR and machine learning literature, to minimise the likelihood of setting an algorithm's parameters to values that lead to a large generalisation error. It requires that we define a resampling method to estimate the generalisation error, and a metric to measure it. We chose leave-one-subject-out cross-validation (LOSO CV) as our resampling method, and the MAE as our metric. The MAE for a classification problem with k classes and N instances is given by

$$\text{MAE}(\mathbf{Y}, \hat{\mathbf{P}}) = \frac{1}{k} \sum_{j}^{k} \frac{1}{N} \sum_{i}^{N} |y_{ij} - \hat{p}_{ij}|. \tag{3.1}$$

where $\hat{p}_{ij}$ denotes the predicted probability that instance i corresponds to class j and y_{ij} that instance's true class, such that $y_{ij} = 1$ if instance i has class j and $y_{ij} = 0$ otherwise. We chose the MAE because it has been shown, albeit only for the binary case, to be the most appropriate metric if the operating conditions (prior class distributions and misclassification costs) are not fully known in advance [HFF12]. Finally, we have to specify the parameters and corresponding values that should be searched by the tuning procedure.

The procedure begins by randomly splitting the data into two sets, approximately 80% (9 users) for training (and validation), and 20% (2 users) for testing. Then, the train and validation errors are calculated via the chosen resampling method and metric on the training data. The estimated test error is then used to choose the best parameter settings following a minimax approach which selects the settings with the lowest upper 95% confidence interval (C.I.). These settings are then used to train the algorithm on the full training set, and calculate its MAE on the test set. This quantity is then compared against the C.I. to validate the parameter settings for the algorithm. If the test error lies within the C.I. of the validation error from the previous step, then we are satisfied that the algorithm is unlikely to over- or under-fit to the data with these parameters. If the test error is above the

upper end of the C.I., then this is evidence that the model did overfit to the training data and should probably be constrained to be less flexible. If, on the other hand, the test error falls below the lower end of the C.I., then the model might have been underfit and benefit from additional flexibility. In either case, we would have to re-assess the tuning procedure, at least for the affected algorithms. Fortunately, this was not the case as all algorithms' validation errors fell within their test C.I.s.

3.1.5 Sensor and Feature Selection

In addition to the accelerometer and gyroscope features described above, we enriched the data-set with features extracted from the pressure signal. Because atmospheric pressure is directly related to altitude, the collected signal is bound to separate some activities with high accuracy at the place where the data were collected—but the generalisation fails if the sensor is moved to a different altitude. In order to avoid potential biases, we first calculated the mean pressure over all samples of the "standing" position in the data-set and subtracted it from the original pressure signal. Then, the pressure signal was subjected to the same pre-processing and feature extraction—with exception of the pairwise correlation feature—procedure as the raw gyroscope signal in the previous experiment.

Using these data, we conducted three additional experiments. In each of these, we estimated the generalisation error for the three classifiers using the same algorithm parameters and estimation procedure as before. In the first experiment we compared the predictive value of different sensor combinations by evaluating all possible combinations via LOSO CV. The best combination was then used to run the second and third experiment, each of which evaluated (via LOSO CV) a different method for reducing the dimensionality of the HAR inference problem. In one of them we applied principal component analysis (PCA) for dimensionality reduction, retaining only the number of principal components (PCs) required to explain 10%, 30%, 50%, 70%, and 90% of the total variance. In the other, the K-W test was applied for feature selection.

The K-W test is a non-parametric statistical test against the null hypothesis that the tested samples were generated by the same distribution. We leveraged the K-W test for supervised feature selection by applying it to each of the features—partitioned into 17 disjoint samples according to the target class for this purpose—in turn. Then, the features were ranked according to the K-W test statistic, and only their top 10^{th}, 30^{th}, 50^{th}, 70^{th}, and 90^{th} percentile retained as inputs for the inference algorithm.

3.2 Results and Discussion

For kNN, the tuning procedure tried the values 2, 5, 10, 20, 40, 80, and 160 for k, the size of the neighbourhood, and considered both weighted (by the inverse distance) and unweighted voting. It lead to $k = 2$ and weighted voting regardless of the problem. For SVM, the procedure tried 10^{-9}, 10^{-6}, 0.001, 1, and 1000 for γ, the Radial Basis Function kernel coefficient, and 0.01, 1.778, 316, 56 234, and 10^7 for C, the penalty term. It lead to $\gamma = 0.001$ regardless of the problem, and to $C = 10^7$ for the move-type and move-type/lie problem, $C = 316.228$ for the 17-activity problem, and $C = 1.778$ for fall detection.

The GBT algorithm, as an ensemble of trees, depends on a tree induction algorithm. We use the Classification and Regression Tree (CART) algorithm [HTF09, ch. 9] for this purpose. Other tree-induction algorithms, namely C4.5 and C5.0, exist, but if trees are shallow (a basic assumption in boosted ensembles), their ability to prune trees is unlikely to make much impact. Trees are kept simple by imposing a maximum of 16 leafs per tree, nine features per split, and a minimum of eleven samples per leaf. To further safeguard against overfitting, each tree is restricted to a 30% sample of the training data. For tuning the GBT, our procedure tried the values 0.02, 0.04, 0.06, 0.08, and 1.0 for α, the learning rate, and 50–1600 for M, the number of boosting iterations. It lead to $\alpha = 0.02$ for the fall detection and the 17-activity, $\alpha = 0.1$ for the move-type and move-type/lie problem, and $M = 1600$ regardless of the problem. We found, however, that the loss gradient flattens considerably at about 200 iterations. The improvements for $M > 600$ are marginal at best, and no longer justify the additional computing time. Because of this we used

750 iterations to train the final GBT.

Table 3.1: MAE and overall Accuracy (%): 17-activities

	LOSO CV			CV		
	GBT	SVM	kNN	GBT	SVM	kNN
All 4s	4.04	4.74	5.99	0.26	1.60	2.51
Crawl H & K	1.69	1.43	1.88	0.10	0.27	0.25
Crawl Mil.	2.12	1.74	2.17	0.19	0.39	0.41
Crouch	6.16	5.77	7.22	0.33	2.57	3.42
Duck walk	1.59	1.27	1.04	0.08	0.23	0.11
Fall	0.21	0.25	0.10	0.05	0.10	0.02
Jump off	0.81	1.06	0.84	0.35	0.48	0.26
Jump on	0.60	0.76	0.58	0.23	0.29	0.13
Lie	2.65	3.24	3.82	0.10	1.05	1.24
Run	5.18	5.38	5.58	1.41	1.84	1.26
Run down	4.05	4.21	4.44	1.08	1.52	1.31
Run up	3.24	3.98	4.39	1.00	1.22	0.94
Sit	6.84	8.47	9.74	0.30	4.25	4.35
Stand	8.53	10.45	11.76	0.43	5.55	5.57
Walk	4.78	4.80	6.03	0.71	1.25	1.53
Walk down	6.07	5.17	7.54	1.24	1.72	2.33
Walk up	3.78	3.89	6.01	0.61	0.91	1.17
Mean MAE	3.67	3.92	4.65	0.50	1.48	1.58
Accuracy	73.29	72.46	61.49	97.68	93.23	88.77

Using these parameter settings, we estimated the subject-dependent and -independent performance for each algorithm via eleven-fold and LOSO CV across all eleven users, respectively. The results for the three multi-class problems are listed in Tables 3.1–3.3. If we look at these results in combination, we note that all three classifiers are able to discriminate among the targeted activities accurately, with the class-wise subject-dependent MAE ranging from 0.02% to 5.57%, and the subject-independent MAE from 0.1% to 11.76%. GBT performs the best on the three multi-class problems, followed closely by SVM. Despite the small difference (0.2 to 1 percentage points) between the two algorithm's average scores, GBT achieves class-wise MAEs that are more evenly distributed across the target classes than SVM.

Table 3.2: MAE and overall Accuracy (%): move-type/lie

	LOSO CV			CV		
	GBT	SVM	kNN	GBT	SVM	kNN
Crawl	1.48	1.10	2.00	0.10	0.23	0.33
Duck walk	1.51	0.92	1.04	0.06	0.16	0.11
Fall	0.17	0.17	0.10	0.05	0.07	0.02
Jump	0.75	0.80	0.83	0.29	0.41	0.27
Lie	3.10	3.45	3.82	0.11	1.19	1.24
Run	3.84	5.76	6.77	1.02	2.05	1.98
Static	3.81	4.49	5.34	0.15	1.32	1.71
Walk	4.18	5.89	7.45	0.87	1.85	2.31
Mean MAE	2.35	2.82	3.42	0.33	0.91	1.00
Accuracy	90.99	90.80	86.74	98.90	97.92	96.79

Table 3.3: MAE and overall Accuracy (%): Move-type

	LOSO CV			CV		
	GBT	SVM	kNN	GBT	SVM	kNN
Crawl	1.38	1.08	2.00	0.10	0.22	0.33
Duck walk	1.48	0.91	1.04	0.06	0.15	0.11
Fall	0.16	0.17	0.10	0.04	0.07	0.02
Jump	0.75	0.79	0.83	0.29	0.40	0.27
Run	4.00	5.74	6.77	0.99	2.04	1.98
Static	0.71	1.10	1.65	0.04	0.18	0.51
Walk	4.35	5.79	7.45	0.85	1.83	2.31
Mean MAE	1.83	2.23	2.83	0.34	0.70	0.79
Accuracy	94.03	93.86	90.41	99.02	98.62	97.74

For the fall detection problem, the results are as follows: the subject-dependent MAE (and Accuracy) scores for GBT, SVM and kNN are 0.06% (99.96%), 0.05% (99.98%), and 0.02% (99.99%), respectively. The corresponding subject-independent scores are 0.17% (99.86%), 0.12% (99.92%), and 0.1% (99.92%). Here, kNN outperforms SVMs by about 0.02%, and GBTs by about 0.05%. Falls, however, are difficult to simulate, and this may be an artefact of the experimental design. Finally, we note that the difficulties in accurately estimating a deployed HAR system's performance from laboratory data are well known, and this system is unlikely to be an exception.

3.2.1 Sensor and Feature Selection

In this subsection, we present and discuss the results from the three sensor and feature selection experiments we described in subsection 3.1.5. Tables 3.4–3.6 each list the MAE and its standard error (SE)—calculated across the eleven folds of the LOSO CV and subsequently averaged over the 17 target classes—as well as the standard deviation (SD) among the target classes from one of the three experiments. The MAE estimates (with precision SE) the generalisation error we can expect on data from unseen individuals, while the SD serves as a measure of how much the MAE varies among the 17 target classes. Each of the entries in these tables summarises a set of class-wise MAEs. Three examples of these are shown in Table 3.7 which lists the class-wise MAEs that result when using the K-W test to select feature subsets of varying sizes as input to the GBT. The results for each combination of the (A)ccelerometer, (G)yroscope, and (P)ressure sensor are given in Table 3.4. The results from the PCA experiments are shown in Table 3.5, where the percentage of the total variance explained is given by the first column and the corresponding number of components (n) by the second. The results from our experiments with K-W feature selection are given in Table 3.6, where the percentile (PCTL) that is being retained is given by the first column and the corresponding number of features (n) by the second.

According to the results shown in Table 3.4, the best combination is indeed that which includes all three sensors (72 features), where the best performance

Table 3.4: MAE (± SE) and SD (all in %) for all Sensor combinations

	GBT		SVM		kNN	
	MAE	SD	MAE	SD	MAE	SD
A G P	3.6 ± 0.9	2.3	3.8 ± 0.8	2.0	4.2 ± 0.8	2.6
A G	3.7 ± 0.9	2.2	3.9 ± 0.8	2.1	4.2 ± 0.8	2.5
A	4.3 ± 0.9	2.3	4.6 ± 0.9	2.2	5.0 ± 0.9	2.5
A P	4.3 ± 1.0	2.3	4.6 ± 1.0	2.2	5.0 ± 1.0	2.6
G P	5.8 ± 1.0	3.3	6.1 ± 1.0	2.7	6.4 ± 1.1	3.1
G	6.1 ± 1.0	2.9	6.5 ± 1.0	2.5	6.5 ± 1.0	2.9
P	10.4 ± 1.8	6.0	10.5 ± 1.2	4.9	10.5 ± 1.6	5.2

Table 3.5: MAE (± SE) and SD (all in %) when using PCA

		GBT		SVM		kNN	
%	n	MAE	SD	MAE	SD	MAE	SD
10	1	9.4 ± 1.1	3.9	9.5 ± 1.1	3.9	9.4 ± 1.1	3.8
30	3	8.0 ± 1.2	3.2	8.4 ± 1.1	3.2	8.1 ± 1.1	3.1
50	9	4.7 ± 0.9	2.7	5.0 ± 0.9	2.5	4.9 ± 1.0	2.6
70	21	4.5 ± 0.8	2.7	4.5 ± 0.8	2.5	4.7 ± 0.9	2.7
90	40	4.4 ± 0.8	2.6	4.1 ± 0.8	2.3	4.4 ± 0.8	2.6

Table 3.6: MAE (± SE) and SD (all in %) with K-W feature selection

		GBT		SVM		kNN	
%	n	MAE	SD	MAE	SD	MAE	SD
10	7	6.6 ± 1.0	2.8	7.1 ± 1.0	2.9	6.6 ± 1.0	2.8
30	21	4.8 ± 1.1	2.5	4.8 ± 0.9	2.6	5.2 ± 1.2	2.7
50	36	3.5 ± 0.9	2.1	3.9 ± 0.9	2.2	3.8 ± 0.8	2.2
70	50	3.5 ± 0.9	2.2	3.8 ± 0.9	2.2	3.7 ± 0.8	2.2
90	64	3.5 ± 0.8	2.3	3.7 ± 0.8	2.0	3.9 ± 0.8	2.4

(MAE: 3.6% ± 0.9) is achieved with the GBT. However, comparable (MAE: 4.3% ± 1) performance can be obtained using only one sensor, namely the accelerometer; thus retaining 40 of the 72 features and reducing the dimensionality by 44%. In contrast, neither the gyroscope (25 features) nor the pressure sensor (7 features) appears to be useful on its own. A particularly bad choice for a single-sensor HAR system is the pressure sensor, especially considering that a dummy model which makes predictions solely based on the class proportions achieves a MAE of 10.3%. The best two-sensor combination is clearly that of accelerometer and gyroscope ($A\ G$), whose performance is close to that of the $A\ G\ P$ combination whilst corresponding to 65 (86%) of the 72 features. Furthermore, while differences among classifiers that are based on the same sensor combination are well below any of their underlying estimates' precision, there is a visible gap separating combinations that include the accelerometer from those that do not.

The results from our PCA experiments in Table 3.5 show that it can maintain a MAE below 5%, while reducing the dimensionality of the three-sensor ($A\ G\ P$) inference problem beyond what is feasible by simply discarding sensors. An average MAE of 4.7% is obtained with only 9 PCs (explaining 10% of the total variance), but even retaining as many as 40 PCs (explaining 90% of the total variance) the performance does not approach that of the $A\ G\ P$, or even the $A\ G$ (65 features), combination in Table 3.4. This can be due to non-linear relationships among features that classifiers can exploit, but linear methods, such as PCA, cannot capture.

As the K-W feature selection experiments in Table 3.6 show, we can improve, albeit only marginally, on the best combination from Table 3.4 if we retain as few as half of the features, thereby halving the inference problem's dimensionality from 72 to 36 features and—assuming the algorithm's time complexity is linear or worse in the number of features—at least halving the run-time. If we decrease the number of features further, we observe deteriorating performance, as expected, for all three algorithms—most notable in the case of kNN—and we might expect that moving in the other direction and increasing the number of features would have the opposite effect, namely to improve performance. However, our data show that this is not necessarily

Table 3.7: MAE (± SE) for GBT with K-W, retaining different percentiles

	30^{th} PCTL	50^{th} PCTL	70^{th} PCTL
All 4s	6.0 ± 2.4	6.0 ± 2.4	6.0 ± 2.4
Crawl H & K	2.3 ± 0.9	1.7 ± 0.5	1.6 ± 0.5
Crawl M	2.7 ± 0.8	2.0 ± 0.5	1.8 ± 0.5
Crouch	5.2 ± 0.8	5.1 ± 0.8	5.2 ± 0.8
Duck walk	1.6 ± 0.5	1.5 ± 0.5	1.2 ± 0.5
Fall	0.8 ± 0.2	0.8 ± 0.2	0.7 ± 0.2
Jump off	1.9 ± 0.5	1.7 ± 0.5	1.7 ± 0.5
Jump on	1.9 ± 0.4	1.7 ± 0.5	1.6 ± 0.5
Lie	4.8 ± 2.4	4.3 ± 2.4	4.3 ± 2.4
Run	7.1 ± 1.6	3.6 ± 1.1	3.3 ± 1.0
Run down	6.8 ± 0.9	3.7 ± 0.8	3.8 ± 0.7
Run up	7.7 ± 1.4	2.1 ± 0.6	2.0 ± 0.5
Sit	7.1 ± 1.3	6.4 ± 1.2	6.5 ± 1.2
Stand	8.6 ± 1.6	8.4 ± 1.4	8.5 ± 1.4
Walk	3.9 ± 0.8	3.0 ± 0.7	3.1 ± 0.7
Walk down	7.8 ± 1.0	4.7 ± 0.7	4.7 ± 0.7
Walk up	5.5 ± 0.7	2.9 ± 0.4	2.8 ± 0.4
Average	4.8 ± 1.1	3.5 ± 0.9	3.5 ± 0.9

the case. While SVM performance improves marginally—starting with a MAE of 3.9% when using 50%, to 3.8% when using 70%, to 3.7% when using 90% of the features—GBT, instead, maintains a stable MAE of 3.5%, regardless if 50%, 70%, or 80% of the features are being retained; and kNN achieves its best performance when using 70% of the features—with larger percentages leading to worse performance.

The class-wise MAEs shown in Table 3.7 illustrate, using the GBT results as an example, what happens when the percentile of features that is retained increases. Note how the reduction of the average MAE when moving from the 30$^{\text{th}}$ to the 50$^{\text{th}}$ percentile can be attributed mainly to the significant reduction from the three running, the "Walk down" and "Walk up" activities, and—to a lesser extent—the "Walk" (horizontally) activity. Hence, at least some of the features that are in the 50$^{\text{th}}$, but not in the 30$^{\text{th}}$ percentile, are useful for discriminating among these activities, and there is little benefit from using more than 30% for applications such as fall detection, where fine-grained distinctions like these have little practical impact.

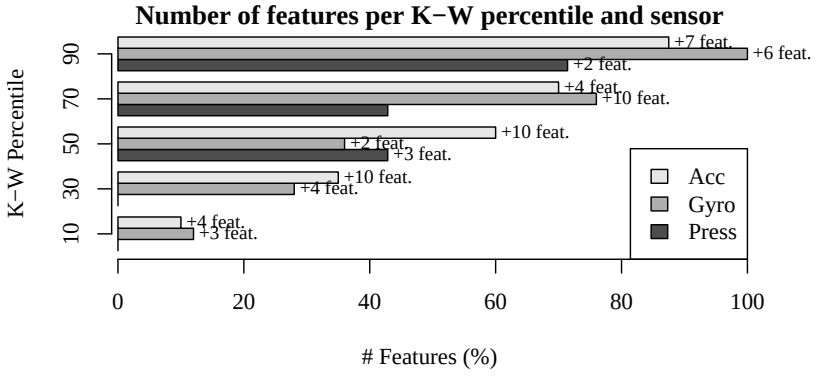


Figure 3.4: Number of features per K-W percentile and sensor, where 100% = all features from that sensor.

We conclude with a summary of the K-W ranked percentiles illustrated in Figure 3.4. The 10$^{\text{th}}$ percentile, amounting to 10% of all the accelerometer, and 12% of all the gyroscope features, consists of the SD of the x and y, and the inter-quartile range of the accelerometer y axes; as well as the SD of the x and y, and inter-quartile range of the gyroscope x axes. The 30$^{\text{th}}$ percentile adds the inter-quartile range, SD, and signal magnitude area features amounting

to 27% of the accelerometer, and 18% of the Gyro features not present in the 10^{th} percentile. The 50^{th} percentile contains all types of features that had been extracted, except pairwise correlations, adding 38% of the accelerometer, 11% of the gyroscope, and 68% of the pressure features not present in the 30^{th} percentile. The 70^{th} percentile adds 25% of the accelerometer, and 63% of the gyroscope features not present in the 50^{th} percentile. The 90^{th} percentile adds what are mostly peak power frequency, spectral entropy, and pairwise correlation features, amounting to 58% of the accelerometer, 100% of the gyroscope, and 50% of the pressure features that were not present in the 70^{th} percentile.

3.3 Conclusions

In this chapter, we tuned and evaluated four machine learning algorithms, viz. SVM, kNN, and GBT on a 17-class HAR problem in the context of an emergency first responder monitoring system. To address at least some of the uncertainty about what the appropriate activities of interest are, we merged the 17 fine-grained activities into more general groups to obtain a total of four (three multi-class and one binary) HAR problems. These classification problems were used to tune, evaluate, and compare the learning algorithms. Our results show that our GBT outperforms the other algorithms on all but one of these four problems. On the fall detection problem, SVM beats our GBT by 0.01 and 0.05 percentage points on the subject-dependent and -independent MAE, respectively. On the full 17-class HAR problem, our GBT achieves subject-dependent and -independent accuracies of over 97% and 73%, respectively. Our results further show that GBT tends to fewer misclassifications, distributed more evenly among the target classes, than kNN or SVM.

Our sensor and feature selection experiments showed that the best among the three evaluated sensors for HAR is the accelerometer, resulting in a MAE of 4.3% ± 0.9 when used with our GBT. At the other extreme we found the pressure sensor, which resulted in a MAE of 10.4%, no better than what we would get when merely guessing the proportion of activities (classes) in the

data-set. The sensor combination that achieved the best results was that with accelerometer, gyroscope, and pressure, with a MAE of 3.6% ± 0.9, closely followed by the accelerometer/gyroscope combination with a MAE of 3.7% ± 0.9. Moreover, our results showed that a simple univariate feature selection method such as the Kruskal-Wallis test can be used to reduce the complexity of a HAR inference problem by as much as 50% while not only maintaining, but even improving the performance of HAR inference algorithms.

In scenarios with a small, clearly defined, and stable population of users, such as the first response teams discussed in this chapter, it is probably more appropriate to focus on the subject-dependent performance. Of course, whether or not this is true depends primarily on whether or not it is feasible to acquire labelled training data from each user. The response team leaders we have spoken to were not opposed to the idea, adding that they had plenty of opportunity to annotate their team's activities during one of the many training sessions that they supervised. Usually, every first responder has their own personal equipment—which would include the wearable sensor board—and it is their responsibility to keep it in order. This begs the question whether we could achieve even better subject-dependent performance if we trained a specific micro model for each user, and used it to make predictions for all instances from that user's IMU, rather than training a single monolithic model with all the data and using that to make predictions for any user, regardless of their identity.

Chapter 4

Subject-Dependent and -Independent Human Activity Recognition with Micro and Macro Models

In this chapter[1], we address the question whether human activity recognition (HAR) micro models, which are personalised to a specific user, might be a way to boost the subject-dependent performance beyond that achieved with the standard macro model approach. The literature discussed in subsections 2.1.2 and 2.1.3 establishes a clear tendency towards better subject-dependent than -independent performance. However, our literature review also shows that the discrepancy is not as clear-cut as we might expect. Moreover, only a few papers directly compare the subject-dependent and -independent HAR performance, making it difficult to quantify the discrepancy. Even fewer papers consider both person-specific micro models and person-independent macro models, and only three [WL12; BG17; Fer+20] report both the subject-dependent performance of micro models and the corresponding macro model, as well as the subject-independent performance of the macro model. Unfortunately, as

[1]The material in this chapter has been published under the title "Comparing Person-Specific and -Independent Models on Subject-Dependent and -Independent Human Activity Recognition Performance" [Sch+20b], which is itself an extended version of [Sch+19].

we have seen, these papers present contradictory results. Two of them [WL12; BG17] find that micro models outperform the corresponding macro model, while the third [Fer+20] comes to the opposite conclusion. Furthermore, all of these experiments are underpowered in terms of the number of data-sets or learning algorithms, particularly given that they report significant variance in the subject-dependent performance between data-sets and learning algorithms. This variance is also the main reason that it is unlikely that combining results from multiple papers in a meta-analysis will yield accurate estimates of the mathematical relationships between the subject-dependent and -independent performance of micro and macro models. Thus, the question whether person-specific micro models are a better approach to subject-dependent HAR than the customary person-independent macro model remains an open one.

In this chapter, we assess whether person-specific models (PSMs) really are an effective way for improving the subject-dependent HAR performance beyond that achieved by the corresponding person-independent model (PIM). A PSM is a predictive model that is trained with data from a specific user. Whenever a new instance is presented to the system, the user's PSM is used to make predictions. The PIM that corresponds to a user population's PSMs is a monolithic model trained with data from all the PSM users. Whenever a new instance is presented to the system, the PIM is used to make predictions, independently of who the prediction is for. In addition to PIMs and PSMs, we also consider three different ensembles of person-specific models (EPSMs). The advantage of EPSMs is that they can exploit data from new users when they become available without having to resort to the data-set that was used to develop the initial model. To do so, we simply fit a PSM for each of the new users and add them to the ensemble. This is not the case for a PIM, which requires the data-set that was used to develop the initial model to exploit data from new users.

Our estimates of the relationships between the subject-dependent and -independent performance achieved with a PIM or PSM should also prove useful for assessing attempts at using limited data from (new or existing) users to create personalised HAR models or personalise existing HAR models. Arguably, such methods should at least exceed the subject-independent or

subject-dependent performance—depending on whether or not the proposed method has access to data from the target user—achieved by a baseline PIM trained with all the data that the final personalised model had access to. This is a reasonable baseline because it demonstrates what can be achieved if a standard, non-personalised, HAR approach is applied to the same data that the personalisation method has access to. They should furthermore aim to meet or exceed the subject-dependent performance achieved by PSMs. Evaluating both PIMs and PSMs can be too onerous and time-consuming in the early development stages of a new personalisation method, when we are still exploring and rapidly prototyping ideas. Our estimates of the relationships between PIMs and PSMs, and their subject-dependent and -independent performances can be used to estimate these, given an estimate of the subject-dependent or subject-independent performance achieved with either PSMs or a PIM.

This chapter proceeds as follows. Section 4.1 describes our methods, including the benchmark data-sets, data pre-processing and segmentation, feature extraction, and activity inference and evaluation. Section 4.2 presents the results of our experiments and their analysis. Section 4.3 discusses our findings in the context of the related literature, and section 4.4 concludes the chapter.

4.1 Methods

We estimate and compare the performance of four machine learning algorithms—L_2-regularised (Ridge) logistic regression, k-Nearest Neighbours (kNN), Support Vector Machine (SVM), and our gradient boosted ensemble of decision trees (GBT)—using a set of features extracted from eight HAR data-sets, which are summarised in Table 4.1. For each data-set, the table cites the relevant publication, lists the number of activities (act) and people (ind), the sampling frequency (Hz), and the average number of trials per activity ± standard error (SE). We chose data-sets that were acquired with wearable inertial measurement units (IMUs) comprised of an acceleration and angular velocity sensor, and worn either on the chest or the wrist. Where

sensors were worn on both wrists we chose the one associated with the right wrist. Unfortunately, the information about whether a user is right- or left handed is unavailable for most data-sets, making it impossible to choose the dominant wrist consistently. All data-sets, except REALWORLD and SAFESENS which only used a chest-worn sensor, used a wrist-worn sensor, and only two data-sets—PAMAP2 and SIMFALL—employed both a wrist- and a chest-worn sensor. Figure 4.1 illustrates how the instances—each of which corresponds to the features extracted from one window—are distributed among the activities. Note that instead of distinguishing falls from activities of daily living (ADLs) in the SIMFALL data-set, which [ÖB14] were able to do with Sensitivity, Specificity, and Accuracy all >99%, we focus on the 16 ADLs shown in the figure. Most of the activity labels are self-explanatory, but some of the activities in the UTSMOKE data-set merit further explanation. "SmokeST" denotes "Smoke Sitting"—smoking (presumably a cigarette) while sitting down—while "SmokeSD" denotes "Smoke Standing"—smoking while standing up. Similarly, "DrinkST" and "DrinkSD" denote "Drink Sitting" (drinking while sitting down) and "Drink Standing" (drinking while standing up), respectively.

Table 4.1: Number of (act)ivities and (ind)ividuals, trials/activity (± SE), and sampling frequency (Hz) for each of the data-sets.

	dataset	act	ind	trials/act	Hz
[Sho+14]	FUSION	7	10	90 ± 0	50
[Bañ+14]	MHEALTH	11	10	38 ± 0	50
[Cha+13]	OPPORT	4	4	590 ± 258	30
[RS12]	PAMAP2	12	9	81 ± 8	100
[SS16]	REALWORLD	8	15	318 ± 42	50
[Sch+17a]	SAFESENS	17	11	91 ± 13	33
[ÖB14]	SIMFALL	16	17	128 ± 8	25
[Sho+16]	UTSMOKE	7	11	859 ± 7	50

The various cross-validation strategies, machine learning algorithms, and calculation of performance measures were implemented in Python (version 3.7.3), using the *scipy* [JOP+01, version 1.1.0], *numpy* [WCV11, version 1.16.2], *pandas* [McK10, version 0.23.3], and *sklearn* [Ped+11, version 0.20.2]

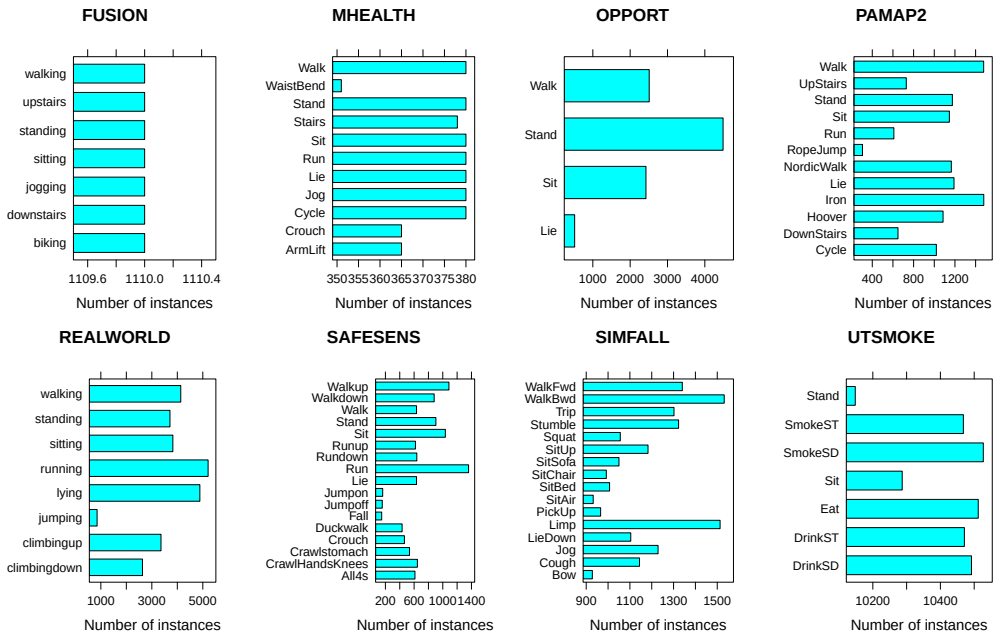


Figure 4.1: Number of instances per activity for each data-set

libraries, and parallelised via GNU parallel [Tan11, version 20161222]. Analysis—t-tests, mixed effects models, and estimated marginal means—and all visualisations were implemented in *R* [RCT19, version 3.6.1], where we used the mixed effects models implementation from the *lme4* library by Bates, Mächler, Bolker, and Walker [Bat+15, version 1.1], and the estimated marginal means implementation from the *emmeans* library by Lenth [Len19, version 1.3.2].

We propose another personalisation-generalisation approach in addition to PIMs and PSMs, which we term an EPSM. An EPSM maintains a PSM for each known user. When an instance for a known user needs to be classified, an EPSM simply applies that user's PSM, but when an instance originates with an unknown user, it applies each user's PSM to obtain confidence scores (e.g., the estimated probability) for each activity of interest. Then the EPSM calculates each activity's mean score, and classifies the instance to the activity with the maximum mean score. To deal with the (very few) users for whom the data do not cover all the activities of interest, and whose PSMs are therefore unaware of some activities and hence unable to generate a confidence score

for those activities, we assume that those activities have a probability of zero. This is not unreasonable if we accept that some people will never perform certain activities (e.g., smoking, military crawling).

This chapter proposes two flavours of *weighted* EPSMs in addition to the basic, unweighted, EPSM described above. A weighted EPSM makes predictions for known users in the same manner as an unweighted EPSM, but when making predictions for unknown users, a weighted EPSM combines its constituent models' predictions via a weighted average. The two types of weighted EPSMs proposed in this chapter differ in how the weights are determined. The first type is the κ-weighted ensemble of person-specific models (WEPSM_κ). A WEPSM_κ weights its constituents' predictions according to each PSM's average κ across all the other training users—i.e., all users except the one whose data are held out for testing and the one whose data (were) used for fitting the PSM. The second type is the baseline-feature-weighted ensemble of person-specific models (WEPSM_{bf}). A WEPSM_{bf} weights its constituents' predictions according to the mean euclidean distance between the PSM (training) user's baseline features and the test user's baseline features. A user's baseline features are the features extracted from an instance (window) of standing or sitting. We obtain our baseline features by sampling, for each user and cross-validation (CV) fold, one instance of each standing and sitting activity. Then, to obtain the weight for a given train- and test-user, we calculate the pairwise distances between each of the two users' instances, and take the mean of the distances to weight the training user's PSM predictions for the test user when aggregating them.

Figure 4.2 provides a graphical overview of the experiment we conducted, whose details are explained in the following subsections. The raw sensor signals from each data-set and sensor are first subjected to pre-processing. The pre-processed data—extracted features and labels, along with metadata identifying the originating data-set, sensor, and user—are then used to evaluate each learning algorithm's ability to infer human activities in terms of subject-dependent (the figure's left-hand branch) and -independent (the right-hand branch) performance. The figure also illustrates the difference between PSMs, unweighted EPSMs, WEPSM_κs, and WEPSM_{bf}s (labelled "WEPSM_BF" in

the figure).

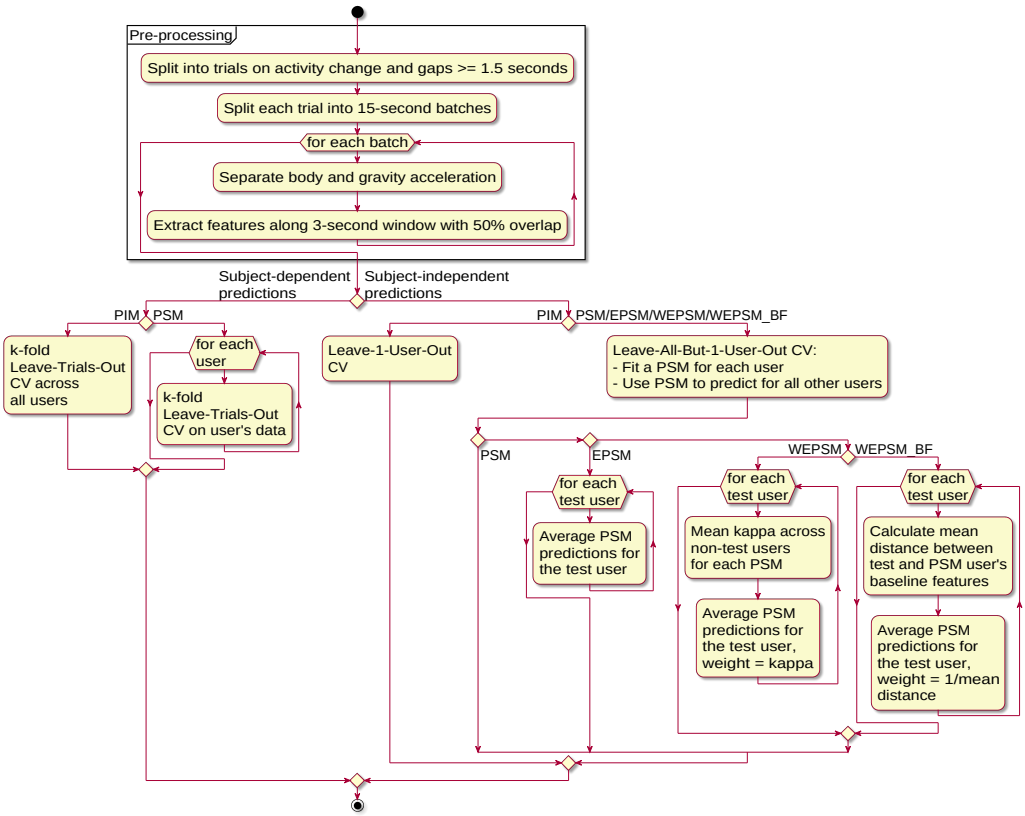


Figure 4.2: Graphical summary of the experiment. Each data-set is pre-processed once, and then used to obtain subject-dependent and -independent predictions with each learning algorithm and personalisation-generalisation approach (PSM, PIM, etc.). Note that an ensemble of PSMs is identical to a PSM in the case of subject-dependent predictions (i.e., for known users).

4.1.1 Pre-Processing, Segmentation, and Feature Extraction

Some data-sets come with a constant timestamp for each trial—presumably introduced when attempting to store POSIX® epoch timestamps in (sub-) millisecond resolution in Microsoft® Excel® spreadsheets. For these data-sets we generate timestamps with a fixed inter-arrival time equal to the data-set's nominal sampling frequency. Then, we (automatically) separate the raw data

into non-overlapping *natural* trials by splitting the signal whenever the activity (label) changes or the inter-arrival time (i.e., the time between two subsequent samples) exceeds 1.5 s. To ensure that we have at least two trials per user and activity, each of the natural trials is then split into non-overlapping batches of 15 seconds. Next, we follow the procedure described in section 3.1. The body and gravity components of each trial's accelerometer signal are separated by an elliptical IIR low pass filter, and a set of time- and frequency-domain features extracted along a sliding 3 s window with 50% (1.5 s). From the angular velocity signal and both acceleration components we extract the mean, standard deviation, skew, and kurtosis, and from the angular velocity and body acceleration signal the spectral power entropy, peak-power frequency, signal magnitude area, and the pairwise correlations between each signal's axes. This amounts to a total of 84 features that are extracted from each window.

4.1.2 Activity Inference and Evaluation

We use logistic ridge regression with $C = 0.98$, a kNN classifier with $k = 2$ and weighted voting, a SVM classifier with a radial basis function with kernel coefficient $\gamma = 0.001$ and cost penalty $C = 316$, and a GBT with learning rate $\alpha = 0.02$ comprised of 750 trees. The parameters for kNN, SVM, and GBT are taken from chapter 3, where we tuned them for subject-independent performance on the 17 activities in our SAFESENS data-set. The ridge parameter of $C = 0.98$ corresponds to weak regularisation, and was chosen to counteract the impact of correlated features. Of course, we ideally would separately tune each algorithm for optimal subject-independent and -dependent performance when used as a PIM and PSM. However, doing so not only would massively increase the complexity of the experiments and analysis of the results, but also raises some technical issues. Such as how one ought to tune PIMs for subject-dependent and PSMs for subject-independent performance. For instance, when optimising a PIM for subject-dependent performance, should we tune the model hyper-parameters for each user separately, or once for all users? Similarly, when optimising a PSM for

subject-independent performance, should we tune its hyper-parameters for each (held-out) user separately, or once across all test users? We decided to avoid both the increased complexity and technical issues, and leave them as a topic of future work. All features are standardised ($[x - \bar{x}]/s$) according to each feature's mean ($\bar{x}$) and (sample) standard deviation (s) in the training data. We use Cohen's Kappa (κ) to quantify the predictive performance because—unlike other performance metrics such as Sensitivity, Specificity, and Accuracy—it corrects for the probability of obtaining the observed level of agreement between the ground truth and predicted labels by chance, and because it is designed to measure predictive performance for multi-class classification.

To estimate an algorithm's subject-dependent performance, the trials are used to generate the folds in a k-fold cross-validation, a method we call Leave-Trials-Out cross-validation [Jor+19]. Leave-trials-out CV ensures that the raw data used to derive an instance in a training split are never used to derive the instances that constitute the corresponding test split, an issue that is bound to occur when working with instances derived from partially overlapping sliding windows [Jor+19], as we do here. PIM performance for known users is estimated by carrying out a k-fold leave-trials-out CV across all the users in each data-set, and PSM performance by carrying out a separate k-fold leave-trials-out CV for each user. In both cases we let $k = n$, where n denotes the number of people in the data-set. To estimate the subject-independent performance, we carry out a leave-m-users-out CV with $m = 1$ for EPSMs and PIMs, and $m = n - 1$ for PSMs.

4.2 Results and Analysis

Figure 4.3 illustrates the trade-off between the subject-dependent performance—i.e., the performance for users who were represented in the data used for training the model—on the horizontal axis, and the subject-independent performance—the performance for users who were not represented in the training data—on the vertical axis. In this figure, each datum corresponds to a single person (user), except in the case of PSMs, where it corresponds to the

median performance a model trained on data from the known user achieved on the other users in the data-set. The symbol and colour indicate which personalisation-generalisation approach (PIM, PSM, EPSM, WEPSM_{κ}, or WEPSM_{bf}) was used. Table 4.2 summarises the results depicted in Figure 4.3, but using the PSM performance for all rather than, as shown in the figure, only that for the average unknown user. The table lists the mean κ (in %) $\pm$ SE for each personalisation-generalisation approach, machine learning algorithm, data-set, and sensor location. To make the results more comparable with other results for these same data-sets, the appendix presents the same tables with the accuracy (Table A.1) and weighted F1-score (Table A.2).

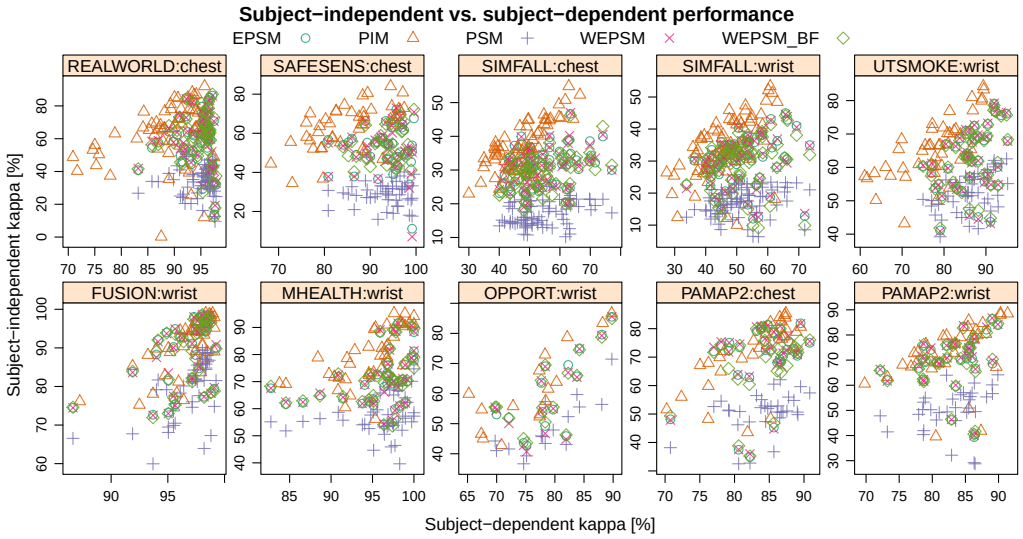


Figure 4.3: Subject-independent (vertical axis) versus subject-dependent (horizontal axis) κ (%) across all learning algorithms. Axes have been scaled to encompass the data for improved visibility. Note how the subject-dependent performance of (E)PSMs tends to be better (further to the right) than that of PIMs, and the clear difference in subject-independent performance between PIMs and PSMs.

Table 4.2: Subject-dependent and -independent κ (%) $\pm$ SE when machine learning algorithms (MLA) are combined with a PIM, a PSM, an unweighted EPSM, a WEPSM_κ, or a WEPSM_{bf}

		subject-dependent		subject-independent				
dataset	MLA	PIM	(E)PSM	PIM	PSM	EPSM	WEPSM_κ	WEPSM_{bf}
FUSION	gbt	97.9 ± 0.3	97.6 ± 0.4	92.4 ± 2.2	81.4 ± 2.1	90.6 ± 2.6	90.7 ± 2.5	90.2 ± 2.6
wrist	knn	94.0 ± 0.9	94.2 ± 1.0	85.9 ± 2.0	74.9 ± 2.6	87.5 ± 2.8	87.5 ± 2.8	87.3 ± 2.8
	glm	96.7 ± 0.6	97.4 ± 0.4	91.9 ± 2.1	79.4 ± 2.7	89.5 ± 2.8	89.5 ± 2.8	89.0 ± 2.7
	svm	98.0 ± 0.3	97.8 ± 0.4	90.9 ± 2.1	80.0 ± 2.3	90.3 ± 2.7	90.4 ± 2.7	90.0 ± 2.6
MHEALTH	gbt	97.5 ± 0.8	97.2 ± 1.2	82.4 ± 3.6	59.5 ± 2.3	72.2 ± 3.5	72.4 ± 3.4	71.5 ± 3.5
wrist	knn	92.9 ± 1.3	93.7 ± 1.4	76.1 ± 3.1	56.0 ± 2.1	71.4 ± 2.6	71.6 ± 2.6	72.8 ± 2.5
	glm	93.2 ± 1.4	95.8 ± 1.4	78.9 ± 3.3	54.1 ± 2.4	70.0 ± 2.9	70.2 ± 2.9	70.8 ± 2.9
	svm	95.5 ± 1.0	96.8 ± 1.0	82.0 ± 2.6	58.0 ± 2.1	72.0 ± 3.9	72.1 ± 3.9	71.4 ± 4.0
OPPORT	gbt	81.5 ± 2.8	83.5 ± 2.4	69.0 ± 7.2	57.9 ± 5.8	66.5 ± 8.1	66.2 ± 8.4	66.7 ± 7.9
wrist	knn	71.1 ± 2.7	74.8 ± 2.7	51.4 ± 4.2	42.9 ± 3.3	54.5 ± 5.4	53.6 ± 5.1	54.8 ± 4.6
	glm	71.9 ± 3.7	76.7 ± 3.0	59.9 ± 7.0	46.2 ± 3.4	56.7 ± 6.7	56.4 ± 7.0	57.2 ± 6.4
	svm	80.3 ± 2.6	81.0 ± 2.4	65.4 ± 6.7	48.4 ± 2.8	61.7 ± 6.6	61.3 ± 7.0	62.3 ± 6.2
PAMAP2	gbt	87.5 ± 0.5	87.7 ± 0.6	77.4 ± 4.3	54.7 ± 2.9	72.4 ± 4.1	72.9 ± 4.2	72.0 ± 3.8
chest	knn	75.5 ± 1.0	78.4 ± 1.2	63.7 ± 2.5	49.0 ± 1.8	67.7 ± 3.2	68.2 ± 3.4	66.3 ± 3.1
	glm	82.5 ± 1.0	85.4 ± 0.9	72.2 ± 3.8	48.8 ± 2.4	69.4 ± 4.9	69.4 ± 4.9	68.6 ± 4.8
	svm	86.0 ± 0.7	85.1 ± 0.8	73.7 ± 4.5	49.7 ± 2.6	69.4 ± 5.1	70.0 ± 5.1	68.5 ± 5.1

Continued on next page

Continued from previous page

		subject-dependent		subject-independent				
dataset	MLA	PIM	(E)PSM	PIM	PSM	EPSM	WEPSM_{κ}	WEPSM_{bf}
PAMAP2	gbt	86.8 ± 1.1	86.0 ± 0.9	78.5 ± 2.8	56.8 ± 2.3	71.7 ± 2.7	72.2 ± 2.6	72.0 ± 2.8
wrist	knn	77.4 ± 1.5	78.9 ± 1.6	65.2 ± 4.1	47.5 ± 2.7	68.1 ± 3.9	68.5 ± 4.0	67.6 ± 3.8
	glm	82.4 ± 1.7	83.5 ± 1.3	74.7 ± 4.1	49.9 ± 3.5	68.8 ± 4.9	69.3 ± 4.9	69.0 ± 4.8
	svm	84.9 ± 1.3	83.3 ± 1.3	73.1 ± 5.1	46.6 ± 3.0	68.3 ± 4.5	68.8 ± 4.5	68.2 ± 4.3
REALWORLD	gbt	93.3 ± 0.6	96.1 ± 0.4	71.7 ± 4.4	37.5 ± 1.9	62.7 ± 4.0	64.1 ± 3.8	63.2 ± 3.9
chest	knn	85.3 ± 1.5	91.3 ± 1.0	59.3 ± 3.4	37.9 ± 2.4	61.8 ± 3.7	62.8 ± 3.6	62.0 ± 3.8
	glm	83.8 ± 1.8	95.4 ± 0.5	60.6 ± 5.8	30.3 ± 2.1	57.1 ± 4.2	58.7 ± 4.2	57.1 ± 4.3
	svm	92.0 ± 0.7	95.5 ± 0.4	62.2 ± 5.1	30.4 ± 2.1	54.8 ± 4.5	56.5 ± 4.3	55.0 ± 4.7
SAFESENS	gbt	93.9 ± 0.9	97.0 ± 0.8	67.6 ± 3.3	27.9 ± 2.0	48.9 ± 4.8	48.7 ± 5.4	53.9 ± 3.3
chest	knn	81.3 ± 1.9	87.8 ± 1.5	55.7 ± 3.5	30.2 ± 1.7	54.7 ± 3.2	54.6 ± 3.3	54.2 ± 2.7
	glm	78.7 ± 1.6	93.1 ± 1.0	64.1 ± 3.0	27.4 ± 1.8	54.0 ± 2.7	53.3 ± 2.7	54.1 ± 2.8
	svm	88.1 ± 1.2	95.2 ± 0.8	66.9 ± 2.7	29.9 ± 1.8	51.9 ± 2.6	53.0 ± 2.4	51.9 ± 2.9
SIMFALL	gbt	57.2 ± 1.2	65.9 ± 1.3	43.9 ± 1.6	19.3 ± 0.7	33.5 ± 1.6	33.5 ± 1.6	32.6 ± 1.5
chest	knn	45.0 ± 0.9	49.5 ± 1.2	30.3 ± 0.7	19.8 ± 0.6	33.3 ± 1.0	33.2 ± 1.1	32.2 ± 0.9
	glm	38.3 ± 0.9	52.3 ± 1.2	34.5 ± 1.1	14.5 ± 0.5	29.1 ± 1.0	29.1 ± 1.0	28.2 ± 1.0
	svm	50.1 ± 0.7	49.5 ± 1.6	38.2 ± 1.4	14.1 ± 0.4	25.9 ± 1.0	26.2 ± 0.9	24.8 ± 1.0
SIMFALL	gbt	55.4 ± 1.5	62.7 ± 1.5	40.8 ± 2.3	19.1 ± 1.0	32.9 ± 2.2	33.0 ± 2.2	32.0 ± 2.2
wrist	knn	44.6 ± 1.2	48.8 ± 1.2	29.2 ± 1.5	19.2 ± 1.0	31.8 ± 1.6	31.9 ± 1.6	31.0 ± 1.6
	glm	37.3 ± 1.4	49.6 ± 1.3	32.7 ± 2.1	16.2 ± 0.9	27.9 ± 1.7	28.4 ± 1.7	27.6 ± 1.8

Continued on next page

Continued from previous page

		subject-dependent		subject-independent				
dataset	MLA	PIM	(E)PSM	PIM	PSM	EPSM	WEPSM_{κ}	WEPSM_{bf}
	svm	48.2 ± 1.3	45.9 ± 1.5	36.0 ± 2.3	13.7 ± 0.7	27.1 ± 1.5	27.3 ± 1.6	26.5 ± 1.7
UTSMOKE	gbt	80.9 ± 1.5	90.8 ± 0.9	68.7 ± 2.9	54.8 ± 1.8	65.4 ± 3.2	65.4 ± 3.3	65.3 ± 3.1
wrist	knn	76.3 ± 1.3	81.2 ± 1.2	61.6 ± 2.4	50.8 ± 1.7	60.7 ± 2.8	60.8 ± 2.9	60.5 ± 2.7
	glm	68.9 ± 2.1	84.1 ± 1.2	63.2 ± 2.5	50.5 ± 1.6	59.4 ± 2.5	59.4 ± 2.5	59.6 ± 2.4
	svm	83.6 ± 1.3	89.1 ± 0.9	69.2 ± 2.7	52.9 ± 1.8	63.6 ± 2.9	63.8 ± 3.0	63.8 ± 2.7

Inspecting these results, it is clear that the subject-independent performance is systematically and substantially worse than the corresponding subject-dependent performance. It is also clear that PSMs perform worse than PIMs in terms of their subject-independent performance. Furthermore, GBT clearly outperforms logistic regression (logreg) and k-Nearest Neighbours (kNN), with few exceptions. The most notable of these is the subject-independent performance of micro models—PSMs, EPSM, WEPSM_{κ}, and WEPSM_{bf}—on the SAFESENS data-set, where both kNN and logistic regression outperform gradient boosted trees, in some cases by over a standard error. However, things are less clear when it comes to comparing GBT and SVM, PIMs and PSMs on subject-dependent performance, PIMs and EPSMs on subject-independent performance, or comparing the different types of EPSMs against each other. To elucidate these matters, and to quantify the obvious differences mentioned above, we turn to statistical analyses which are discussed in the remainder of this section.

4.2.1 Analysis of the Subject-Dependent Performance

We can pair the performance when a PSM is combined with a machine learning algorithm and applied to the data from a known person for a given data-set and sensor, to the performance when the same algorithm is combined with a PIM and applied to the same data-set, sensor, and person. A paired t-test of these data yields a 95% confidence interval (C.I.) of 4.1 to 5.2 percentage points (hereafter, points) for the difference between the κ achieved with PSMs and that achieved with PIM, with a mean difference of 4.6 points ($t_{442} = 16.2$, $\text{P} < 2.2 \times 10^{-16}$), suggesting that we can be 95% confident that a PSM outperforms a PIM on data from known users by 4.1 to 5.2 points. However, it is unlikely that the t-test's underlying assumption of identically and independently distributed (IID) data is met, because the difference in the subject-dependent performance between PIM and PSM might depend not only on the data-set—which is expected due to the different activities of interest, and evident in Figure 4.3—but also on the learning algorithm.

Most of the standard statistical techniques for the analysis of experiments,

such as the venerable t-test or Analysis of Variance (ANOVA), assume that the data (or residuals) are identically and independently distributed (IID). A more appropriate tool for analysing non-IID data is the linear mixed-effects model. Linear mixed effects models extend linear regression with so-called *random effects* which allow us to impose structure on the residuals. We can, for example, specify that the performances within data-sets are correlated, or even that the difference in performance between classifiers varies depending on the data-set. The random effects are assumed to add up to zero, and hence the fixed effects (which are analogous to linear regression coefficients) can be estimated via (restricted) maximum likelihood. A linear mixed-effects model, like the linear regression model it is based on, is built on the assumption of normally distributed residuals. Generalised linear mixed-effects models (GLMMs) extend linear mixed-effects models to non-normal data, analogous to the generalised linear model. Like the generalised linear model, GLMMs employ a link function, such as the logit, and error distribution from the exponential family to model non-normal responses. For a detailed treatment of linear mixed-effects models and GLMMs we refer interested readers to [GH06]. We therefore use logistic GLMMs—a GLMM with a logistic link function and binomially distributed errors—to analyse the subject-dependent and -independent performance, and the relationship between them. We consider the personalisation-generalisation approach (PGA)—e.g., PIM, PSM, EPSM, or WEPSM$_\kappa$—and the machine learning algorithm (MLA) as explanatory variables (fixed effects), and the data-set and sensor as random effects.

We use a GLMM to model the subject-dependent performance as a combination of (fixed) effects for the machine learning algorithm and personalisation-generalisation approach—PIM or PSM, since subject-dependent EPSM performance is identical to that of its constituent PSMs—and a random effect to control for the variation of the personalisation-generalisation approach effect between data-sets. This model explains the observed variation in the response with a residual standard deviation of 1.0157 between data-sets, and with a standard deviation of 0.0255 between data-sets' sensors. This model reveals that the (random) effect of applying PSM varies with a standard deviation of 0.2929 between data-sets, where it is weakly negatively correlated

(−0.11) with PIM performance, and with a standard deviation of 0.0407 between data-sets' sensors, where it is strongly positively correlated (0.75) with PIM performance. This captures the intuition that a PSM confers less advantage on data-sets on which a PIM already performs well. The maximum likelihood estimates of the fixed effects, which are shown in Table 4.3, indicate that GBT—with an estimated κ of 89.4% and a 95% C.I. of 80.6% to 94.4% when used as a PIM—outperforms SVM by 19.5% (18.5% to 20.5%), logistic regression by 45.5% (44.9% to 46.2%), and kNN by 46% (45.3% to 46.6%), regardless of whether they are combined with a PIM or a PSM. They further show that PSMs outperform the corresponding PIM by 43.5% (17% to 76%, P = 0.00058) on subject-dependent performance.

Table 4.3: GLMM estimates (β), 95% C.I.s, and P-values of the fixed effects on subject-dependent performance associated with learning algorithms and personalisation-generalisation approaches.

Coefficient	2.5%	β	97.5%	P
(Intercept)	1.425	2.129	2.833	3.1×10^{-9}
kNN	−0.628	−0.616	−0.604	$<2.0 \times 10^{-16}$
logreg	−0.620	−0.608	−0.596	$<2.0 \times 10^{-16}$
SVM	−0.229	−0.217	−0.204	$<2.0 \times 10^{-16}$
PSM	0.155	0.361	0.566	5.8×10^{-4}

Both the paired t-test and the GLMM analysis indicate that PSMs outperform the corresponding PIM on subject-dependent performance. The GLMM estimates that PSMs outperform PIMs by 17% to 76% (with a mean of 43.5%) on subject-dependent performance, in terms of the odds-ratio, the t-test estimates the difference between 4.1 to 5.2 points, with a mean difference of 4.6 points. These estimates are consistent with each other. The GLMM estimate for GBT with PSM is κ = 92.3% with a C.I. of 85.5% to 96.1%. With PIM it is κ = 89.4% with a C.I. of 80.6% to 94.4%. 89.4% + 4.6 points equates 94%, which is only 1.7 points above the GLMM estimate and well within the C.I. of 85.5% to 96.1% postulated by the GLMM. For kNN, the GLMM estimates the κ achieved with PSM at 86.7% with a C.I. of 76.2% to 93%, while the κ achieved with PIM is estimated at 82% with a C.I. of 69.2%

to 90.2%. 82% + 4.6 equates 86.6%, which not only lies well within the C.I. postulated by the GLMM, but is exceedingly close to the point estimate of 86.7%. Similarly for logistic regression, where the GLMM estimates the κ achieved with PSM at 86.8% with a C.I. of 76.3% to 93.1%, and with PIM at 82.1% with a C.I. of 69.4% to 90.3%. 82.1% + 4.6 equates 86.7%, which is only 0.1 point below the GLMM point estimate (and well withing the C.I. postulated by the GLMM). For SVM, the GLMM estimates the κ achieved with PSM at 90.7% with a C.I. of 82.6% to 95.2%, and the κ achieved with PIM at 87.1% with a C.I. of 77% to 93.2%. 87.1% + 4.6 equates 91.7%, which is only one point above the GLMM point estimate, and well within the 95% C.I. postulated by the GLMM.

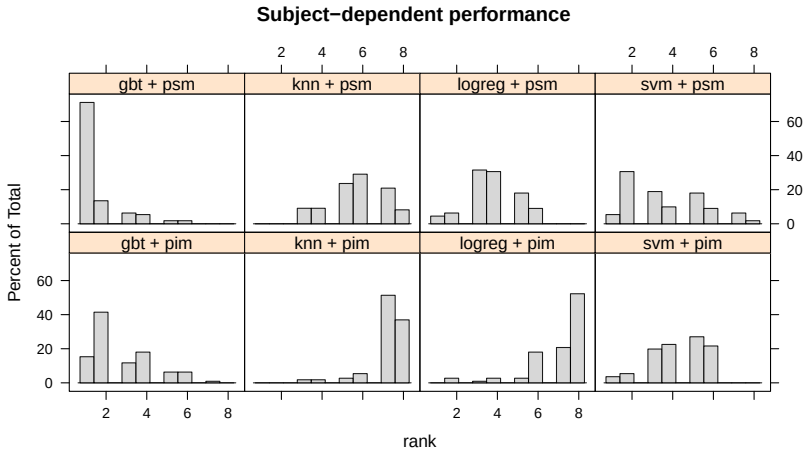


Figure 4.4: Distribution of the user-wise subject-dependent ranks of each learning algorithm + personalisation-generalisation approach. A person-specific gradient boosted GBT (gbt + psm) outperforms the other methods for over 70% of users.

Both the t-test and GLMM rely on statistical assumptions. To compare the evaluated methods without relying on statistical assumptions, we rank the methods by their κ within each user, data-set, and sensor. Figure 4.4 illustrates the distribution of each method's (learning algorithm + PSM or PIM) ranks across all data-sets, sensors, and users. While no single method dominates across all users, the figure shows that PSMs with GBTs perform better than all other methods for nearly 80% of users, and better than all but

one method for over 10% of users, which confirms our statistical analysis. It is also clear that PSMs more often than not outperform a PIM for any given algorithm.

4.2.2 Analysis of the Subject-Independent Performance

A binomial logistic GLMM with fixed effects for the learning algorithm and personalisation-generalisation approach, and a random effect for sensors nested within data-sets explains the variation in subject-independent performance with a residual error that varies with a standard deviation of 0.781 between data-sets and with a standard deviation of 0.0218 between data-sets' sensors. The maximum likelihood estimates of the fixed effects, which are shown in Table 4.4, indicate that GBT—with an estimated κ of 71.3% (59.2% to 81.0%, P = 0.001) when used as a PIM—outperforms kNN by 20.2% (19.7% to 20.6%), logistic regression by 21.7% (21.3% to 22.2%), and SVM by 15.8% (15.3% to 16.3%). PIM outperforms PSM by 55.9% (55.3% to 56.2%), EPSM by 17.5% (17% to 18.1%), WEPSM_κ by 16.4% (15.8% to 16.9%), and WEPSM_{bf} by 18.4% (17.8% to 18.9%). All P-values $< 2 \times 10^{-16}$. This analysis clearly shows that PIM performs better for unknown users (i.e., on subject-independent performance) than the other personalisation-generalisation approaches, and that ensembles of PSMs perform better than a PSM. However, because the fixed effects for the different EPSMs—unweighted (EPSM), WEPSM_κ, or WEPSM_{bf})—estimate the difference between the particular EPSM and PIMs, and because their estimates are quite similar, this analysis on its own cannot compare the different types of EPSMs. To compare the different types of EPSMs we employ paired t-tests and estimated marginal means (also known as least-squares means) analysis.

According to the estimated marginal means, which are shown in Figure 4.5, the odds achieved by WEPSM_κs are 1.4% higher than those achieved by unweighted EPSMs (P = 0.0009), which in turn are 1.1% higher than those achieved by EPSMs weighted by the inverse distance between the train and test user's baseline features (P = 0.0159). A paired t-test of the difference in the subject-independent performance between WEPSM_κ and EPSM yields a

Table 4.4: GLMM estimates (β), 95% C.I.s, and P-values of the fixed effects on subject-independent performance associated with learning algorithms and personalisation-generalisation approaches.

Coefficient	2.5%	β	97.5%	P
(Intercept)	0.367	0.907	1.448	1.0×10^{-3}
kNN	−0.231	−0.225	−0.219	$<2.0 \times 10^{-16}$
logreg	−0.251	−0.245	−0.239	$<2.0 \times 10^{-16}$
SVM	−0.178	−0.172	−0.166	$<2.0 \times 10^{-16}$
PSM	−0.825	−0.818	−0.812	$<2.0 \times 10^{-16}$
EPSM	−0.199	−0.193	−0.186	$<2.0 \times 10^{-16}$
WEPSM	−0.186	−0.179	−0.172	$<2.0 \times 10^{-16}$
WEPSM_{bf}	−0.210	−0.203	−0.196	$<2.0 \times 10^{-16}$

mean difference of 0.32 points with a 95% C.I. of 0.20 to 0.44 points and a t-value of 17 on 443 degrees of freedom, which corresponds to a P-value of 3.48×10^{-7}. This shows that a weighted EPSM significantly (albeit by less than one third of a point) outperforms an unweighted EPSM. A paired t-test of the difference in the subject-independent performance between EPSM and WEPSM_{bf} yields a mean difference of 0.38 points, a 95% C.I. of 0.21 to 0.55 points, and a t-value of 4.45 on 435 degrees of freedom, corresponding to a P-value of 1.1×10^{-5}. This shows that using baseline features for weighting the PSM predictions performs significantly worse (albeit by little more than one third of a point) than an unweighted EPSM. Both the estimated marginal means and paired t-tests lead to the conclusion that WEPSM_{κ}s significantly outperform unweighted EPSMs, which in turn significantly outperform EPSMs that are weighted by the inverse mean distance between the train and test user's baseline features.

Analogous to the subject-dependent ranks shown in Figure 4.4, Figure 4.6 illustrates the distribution of each method's (personalisation-generalisation approach + learning algorithm) subject-independent ranks across all data-sets, sensors, and users. The figure clearly shows that PIM + GBT ranks first or second for over 60% of users—more often than any other method—and third or fourth for over 20% of users, which confirms the statistical analysis's finding that PIM + GBT outperforms other methods on subject-independent



Figure 4.5: Estimated marginal means for comparing the subject-independent performance of personalisation-generalisation approaches across learning algorithms. PIMs clearly outperform the other personalisation-generalisation approaches for any learning algorithm, and ensembles of PSMs (particularly WEPSM$_\kappa$s) clearly outperform its constituent PSMs.

performance. They also show that PIMs tend to perform better than other personalisation-generalisation approaches, in particular PSMs which mostly rank in the bottom third. We can also see how κ-weighted EPSMs tend to shift the rather flat distribution of unweighted EPSMs slightly towards the better (lower) ranks on the left, particularly when combined with kNN or GBT.

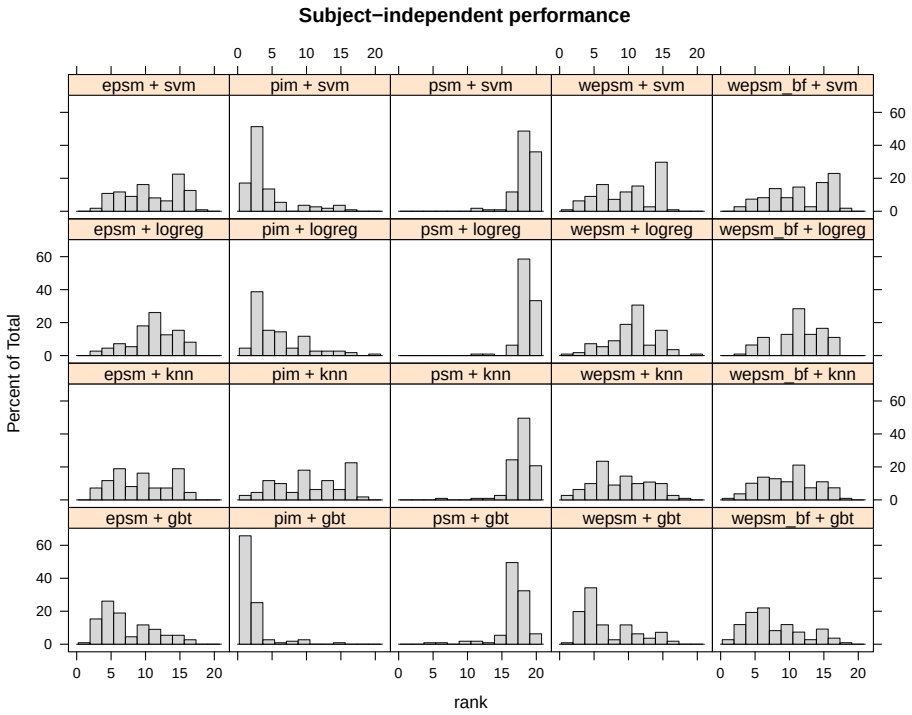


Figure 4.6: Distribution of the user-wise subject-independent ranks of each personalisation-generalisation approach + learning algorithm. PIM + GBT outperforms the other methods for over 60% of users, and performs second-best for over 20% of users.

4.2.3 Comparing Subject-Dependent and Subject-Independent Performance

We use a binomial logistic GLMM with a fixed effect for the performance type (subject-dependent or -independent), one for the personalisation-generalisation approach (PIM, PSM, EPSM, etc.), and one for the interaction between

them. There are two random effects, one for sensors nested within data-sets, and one for the four learning algorithms. The reference level (intercept) corresponds to the subject-dependent performance achieved by PIMs. This model shows that the subject-dependent performance varies with a standard deviation of 0.1770 between learning algorithms, 0.825 between data-sets, and 0.0382 between data-sets' sensors. The model's estimates for the fixed effects are shown in Table 4.5, along with their 95% C.I.s and P-values. According to these estimates, PIMs achieve a subject-dependent κ, averaged over learning algorithms, of 82.2% (71.8% to 89.3%, $P = 4.8 \times 10^{-7}$) and PSMs outperform PIMs by 46% (44.9% to 47.2%, $P < 2 \times 10^{-16}$) in terms of the subject-dependent odds, with an estimated mean κ of 87.1% and a C.I. of 78.9% to 89.3% (according to the estimated marginal means shown in Figure 4.7). The subject-independent odds of PIMs are estimated at 48.1% (47.7% to 48.4%) of their subject-dependent odds, with a κ of 69% and an (estimated marginal means) C.I. of 55.1% to 80.1%. The subject-independent odds of PSMs are 13.6% of their subject-dependent odds, with a κ of 47.9% and an estimated marginal means C.I. of 33.7% to 62.4%. The subject-independent odds of EPSMs are 27% of their subject-dependent odds, with a κ of 64.6% and estimated marginal means C.I. of 50.2% to 76.8%. The subject-independent odds of WEPSM_κs are 27.4% of their subject-dependent odds, with a κ of 64.9% and an estimated marginal means C.I. of 50.6% to 77%. The subject-independent odds of WEPSM_{bf} are 26.8% of their subject-dependent odds, with a κ of 64.4% and an estimated marginal means C.I. of 50% to 76.6%.

4.3 Discussion

Our analysis of the results shows that, on average, the best subject-dependent performance is achieved with PSMs and the best subject-independent performance with a PIM. Hence, in order to simultaneously optimise subject-dependent and -independent performance, we should use a PIM for unknown users and PSMs for known users wherever possible. If we use a PIM, rather than a PSM, to make predictions for known users we forego an expected improvement of over 43% in terms of the odds of a correct classification. For

Table 4.5: GLMM estimates (β), 95% C.I.s, and P-values of the fixed effects associated with subject-independent (SI) performance and personalisation-generalisation approaches

Coefficient	2.5%	β	97.5%	P
(Intercept)	0.930	1.530	2.130	5.8×10^{-7}
PSM/(E)PSM$_{(bf)}$	0.371	0.379	0.387	$<2.0 \times 10^{-16}$
SI	−0.739	−0.732	−0.725	$<2.0 \times 10^{-16}$
SI + EPSM	−0.586	−0.575	−0.564	$<2.0 \times 10^{-16}$
SI + PSM	−1.272	−1.261	−1.250	$<2.0 \times 10^{-16}$
SI + WEPSM	−0.572	−0.561	−0.551	$<2.0 \times 10^{-16}$
SI + WEPSM$_{bf}$	−0.595	−0.584	−0.574	$<2.0 \times 10^{-16}$

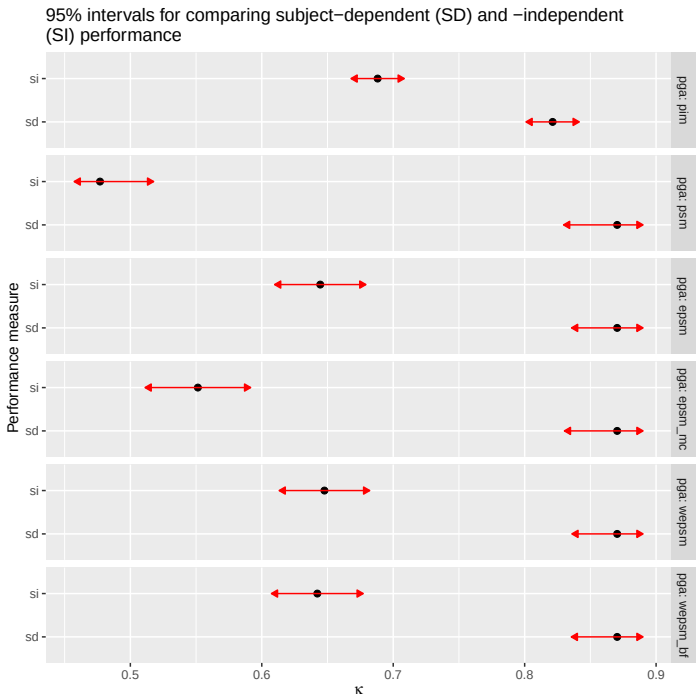


Figure 4.7: Estimated marginal means for the (average) difference between subject-dependent (SD) and -independent (SI) performance across personalisation-generalisation approaches. Subject-dependent performance is clearly better than subject-independent performance, a discrepancy that is minimised with PIMs. The confidence intervals for the subject-independent performance of PIMs and ensembles of PSMs, particularly WEPSM$_{\kappa}$s, overlap.

the data-sets and models investigated in this chapter this corresponds to 4.1 to 5.6 percentage points difference in Cohen's κ. Our analysis also shows that we should expect this discrepancy to be more pronounced if a PIM performs badly.

If, on the other hand, we use a PSM rather than a PIM for unknown users we forego an expected improvement of nearly 56% in terms of the odds of a correct classification. If a PIM is not practicable—e.g., because we do not have access to the original training data when the time comes to integrate new users' data into the HAR model—then an ensemble of PSMs can be employed. Among the three approaches for forming an EPSM which we considered in this chapter, the κ-weighted EPSM emerged as a slightly but significantly better method for forming an EPSM than an unweighted EPSM or a baseline-feature-weighted EPSM. Although a PIM performs significantly better than a WEPSM_κ, the difference in odds is estimated at a mere 16.4%, which is only 0.6 percentage points bigger than the difference between the subject-independent κ of GBT and SVM, the two best learning algorithms in our experiments. Our analysis further shows that GBT significantly outperforms kNN, L_2-regularised logistic regression, and even SVM.

Let us now put these results in context by recalling our review of the state of the art on these data-sets from subsection 2.1.1. The state of the art for the FUSION data-set achieves a subject-dependent accuracy of >99% [JY15]. Our GBT PIM achieves an accuracy of 98.3% ± 0.3 and our GBT PSM an accuracy of 98% ± 0.3 on this data-set. On the OPPORT data-set, the state of the art achieves a subject-dependent error rate of 18% [ZWL15] and F1-score of 93% [OR16; HHP16]. Our GBT PSM achieves a subject-dependent error rate of 10.9% ± 1.5 (GBT PIM: 12.2% ± 1.9) and F1-score of 89.1% ± 1.5 (GBT PIM: 87.8% ± 1.9) on this data-set. According to Jordao, Nazare, Sena, and Schwartz [Jor+19], the state of the art for the MHEALTH data-set achieves a subject-dependent accuracy of 92% (with a C.I. of 88% to 96%), and a subject-independent accuracy of 95% (C.I.: 91% to 98%). Our GBT PIM achieves a subject-dependent accuracy of 97.8% ± 0.7 (GBT PSM: 97.5% ± 1.1), and a subject-independent accuracy of 84% ± 3.3 on this data-set. Following the same paper [Jor+19], the state of the

art for the PAMAP2 data-set achieves a subject-dependent accuracy of 82% (C.I.: 77% to 88%) and a subject-independent accuracy of 85% (C.I.: 76% to 94%). Our GBT PSM achieves a subject-dependent accuracy of 88.9% ± 0.6 (GBT PIM: 88.8% ± 0.4) when using the chest-mounted sensor, and our GBT PIM a subject-independent accuracy of 80.6% ± 2.6 when using the wrist-mounted sensor from this data-set.

On the REALWORLD data-set, the state of the art achieves a subject-dependent accuracy of 89% and F1-score of 90% [SS16]. Our GBT PSM achieves a subject-dependent accuracy of 96.9% ± 0.3 (GBT PIM: 94.5% ± 0.5) and F1-score of 96.8% ± 0.5 (GBT PIM: 94.7% ± 0.5) on this data-set. The state of the art for the SIMFALL data-set achieves a subject-dependent accuracy of 76% [VGR20]. Our GBT PSM achieves a subject-dependent accuracy of 68.1% ± 1.2 (GBT PIM: 60% ± 1.1) when using the chest-mounted sensor and 65.1% ± 1.4 (GBT PIM: 58.3% ± 1.4) when using the wrist-mounted sensor from this data-set. On the UTSMOKE data-set, the state of the art achieves a subject-independent accuracy of 90% [AF18]. Our GBT PIM achieves a subject-independent accuracy of 73.2% ± 2.5 (SVM PIM: 73.6% ± 2.3) on this data-set.

To summarise, our approach performs comparable (i.e., within the margin of error) to the state of the art for all but one (UTSMOKE) data-set in terms of the subject-independent performance. In terms of the subject-dependent performance, our approach performs clearly worse than the state of the art (by about 8 points) on only one data-set (SIMFALL), within one percentage point on another (FUSION), and better than the state of the art on four data-sets—viz. REALWORLD, UTSMOKE, MHEALTH, and PAMAP2. This shows that our gradient boosted ensemble of decision trees, combined with PSMs for subject-dependent or a PIM for subject-independent performance, performs comparable to the state-of-the-art for a wide range of HAR problems, without problem-specific feature engineering or tuning of model hyper-parameters.

4.4 Conclusions

This chapter compared the subject-dependent and -independent performance of PIMs, PSMs, and three types of EPSMs—unweighted, κ-weighted, and baseline-feature-weighted—when combined with four popular HAR algorithms across eight HAR data-sets, seven of which are publicly available. Our analysis with GLMMs shows that our GBT significantly outperforms the other algorithms on both subject-dependent and -independent performance, that PSMs outperform the corresponding PIM by 43.5% (in terms of the odds of correct versus incorrect classification) on subject-dependent performance, and that a PIM outperforms the corresponding PSMs and κ-weighted EPSM by 55.9% and 16.4%, respectively, on subject-independent performance. Furthermore, our analysis of the subject-independent performance shows that WEPSM_{κ}s significantly outperform unweighted EPSMs, albeit by as little as 0.32 percentage points—1.4% in terms of the odds—and that an unweighted EPSM significantly outperforms a WEPSM_{bf} by about the same amount.

Although PIMs outperform EPSMs on subject-independent performance, EPSMs have the advantage that they can easily exploit data from a new user, without accessing any other user's data. To do so, a PSM is fitted to the new user's data and added to the ensemble. A PIM on the other hand needs to be refit to the entire data-set after the new user's data have been added to it. In practice, this means that human sensing system operators that use a PIM approach need to retain all their user data indefinitely if they want to exploit data from new users. This is not particularly desirable. Besides the costs associated with storing and managing any kind of data, storing massive amounts of data about people and their behaviours also comes with real potential for legal issues.

Legally speaking, a human sensing system's operator is the data controller (or at least a data processor) for all the personal data they store. The European Union's General Data Protection Regulation (GDPR) requires that data controllers implement "appropriate technical and organisational measures" to ensure that their users can exercise their data subject rights. Data subject rights include the right to data portability, the right to access

to information, the right of rectification (colloquially known as the right to correction), and the right to erasure (more commonly known as the right to be forgotten). Of course, not all data are personal data, only those "which can be used to identify an individual, natural person." However, given the wide range of data employed for human sensing and the rapid advances in artificial intelligence, it is difficult to predict if and when data which are currently insufficient "to identify an individual, natural person," cease to be insufficient. Either way, the uncertainty about the scope of personal data is in itself motivation to reduce the amount of data about users' behaviours that human sensing systems rely on. EPSMs are one way to achieve this, and our experiments and their analysis provides HAR researchers and practitioners with the estimates needed to make an informed decision about whether the 16% penalty on the subject-independent performance that EPSMs incur is worth the 44% gain in subject-dependent performance and the benefit of not having to store user data.

Our approach—a gradient boosted ensemble of decision trees, combined with person-specific models for known and a person-independent model for unknown users—performs comparable to the state of the art on one data-set and outperforms the state of the art on four data-sets in terms of subject-dependent performance. In terms of the subject-independent performance, our approach performs comparable to the state-of-the-art on all but one of the data-sets for which the subject-independent performance has been published.

Chapter 5

Human Activity Recognition with Expert Hierarchies

In human activity recognition (HAR) applications, it is almost always natural and easy to arrange the activities of interest in a hierarchy, for example by placing the most general categories (e.g., "mobile" and "stationary") at the top or root of the tree, proceeding to increasingly specific categories ("walk" and "run"), and terminating with the most specific categories ("walk upstairs" and "walk downstairs") at the leaves. Furthermore, it is not uncommon that a HAR system's end users find it difficult to precisely specify which activities need to be recognised, let alone the activities' priors and misclassification costs, which are needed to properly tune classifiers. In chapter 3, we partially addressed this, albeit admittedly in a rather ad-hoc fashion, by re-arranging the original seventeen-class activity problem into classification problems with fewer classes. These simplified activity groupings were based on our understanding of which categories firefighters consider the most important to discriminate. In this chapter[1], we take a more systematic approach to exploit that domain knowledge via expert hierarchies, which is the term we use to distinguish nested dichotomies that encode domain knowledge from nested dichotomies constructed by some other process.

[1]The material in this chapter has been published under the title "Using Domain Knowledge for Interpretable and Competitive Multi-class Human Activity Recognition" [Sch+20a].

5. Human Activity Recognition with Expert Hierarchies

As we noted in section 1.1, nested dichotomies are particularly appealing for HAR applications because they make it possible to develop increasingly fine-grained HAR capabilities iteratively. Having a classifier that can accurately distinguish between, for example, stationary and mobile behaviours at an early stage of the development life cycle not only enables early systems-level testing and end user feedback, but can speed up the annotation process—a task which is error-prone and often requires a disproportionate expenditure of human effort—for more specific activities. These advantages have inspired several HAR applications of hierarchical classification [Mat+04; Kar+06; FG15]. Unfortunately, there has been little research into how hierarchical approaches to HAR inference compare to other multi-class decomposition methods, such as one-versus-one (OVO) and one-versus-all (OVA). This is particularly striking because HAR problems tend to be multi-class problems, and because the performance of classification algorithms can be significantly affected by whether and how the multi-class problem is decomposed into a set of binary classification problems. Thus, it is unclear whether or not the benefits of a hierarchical approach for HAR come at the cost of worse predictive performance, and if so, just how high that cost might be.

Hierarchical classification might offer yet another benefit to the HAR community. It might be a way to combine existing HAR data-sets into a single training data-set, and transfer the models trained on it to yet another HAR data-set. One of the main challenges in this undertaking is the fact that few, if any, of the many publicly available HAR data-sets cover exactly the same activities of interest. This can be seen in the data-sets we used in chapter 4. Basic postures and activities such as standing, sitting, and walking are almost always among the activities of interest. But many data-sets include other activities, such as "biking," "jogging," "smoking while sitting" and "smoking while standing," or "eating," which are much less common, some of them even unique to a single data-set. One approach for dealing with this is to simply use only those activities that appear in all data-sets and ignore the rest. Clearly this is not the most efficient use of the data, which, as we have pointed out, are costly to acquire. Another approach is to keep all the activities and turn them into the concepts of a bigger multi-class classification

problem. Expert hierarchies offer a middle path between these two extremes in that they allow us to use all the data while controlling the complexity of the multi-class problem.

In this chapter, we present the first empirical evaluation of the effect that different approaches to multi-class classification have on the performance of various learning algorithms on a multi-class HAR problem. This is also the first direct comparison of hierarchical classification guided by domain knowledge with standard domain-agnostic multi-class approaches on a multi-class HAR problem. We formulate a novel threshold that indicates when a nested dichotomy's branch cannot possibly be on the path to the predicted class, and therefore does not need to be evaluated to predict the most likely activity. Our results show that domain knowledge can be used to construct a multi-class classifier that has lower computational complexity and is easier to interpret than, but performs comparable to OVA, which is the most popular multi-class approach in practice.

The remainder of this chapter is organised as follows. The following section describes the multi-class decomposition methods used in this chapter. Then, in section 5.2 we present a novel shortcut for making predictions with nested dichotomies which is appropriate if we only need to predict the most likely class and not the class-wise probabilities. After that, section 5.3 describes the computational experiments, whose results show that expert hierarchies are able to compete with OVA, and indeed many of the other multi-class decomposition methods discussed in section 5.1, regardless of whether we look at the results for the original multi-class problem or those for the binary classification problem induced by an expert hierarchy's topmost dichotomy. The results, presented in section 5.4, also show that Ensembles of Expert Hierarchies perform comparably to an equally sized Ensemble of Nested Dichotomies on the multi-class problem, but with significantly lower variance among both the cross-validation folds and the learning algorithms than the ensemble of nested dichotomies. Section 5.5 concludes our presentation of the results by summarising and discussing the main findings. Finally, section 5.6 concludes the chapter.

5.1 Multi-Class Decomposition Methods

This section discusses the multi-class decomposition methods that we use in this chapter. We present them in three groups: flat decomposition strategies (5.1.1), strategies based on error-correcting output codes (5.1.2), and hierarchical strategies (5.1.3).

5.1.1 Flat Decomposition Strategies

An intuitive approach for decomposing a multi-class problem into a set of binary classification problems is to use an indicator matrix with one column per class that encodes whether or not an observation belongs to that class. This method is known as one-versus-rest, or OVA, and discussed in more detail by Park [Par12, p. 16]. OVA requires fitting, storing, evaluating, and averaging k models for a k-class problem, one model per class. It is the default method for handling multi-class classification problems in most machine learning libraries and packages, including Weka [Hal+09] and Scikit-learn [Ped+11]. A somewhat more elaborate method has become known as pairwise classification, one-versus-other, or OVO [Fri96; HT98]. OVO fits one model for each pair of classes, using only those observations that belong to either of the two classes. OVO requires fitting, storing, and evaluating $k(k-1)/2$ models, which might explain why, while an implementation is available in most machine learning libraries, it is not the default multi-class decomposition method in any of them. Weka, for example, implements OVO as an option to its `MultiClassClassifier` class, which also implements error-correcting and one-vs-all, with the latter being its default multi-class decomposition method [Bou+16], and Scikit-learn has a `OneVsOneClassifier` which can be used with any classifier conforming to the Scikit-learn API [Bui+13] instead of OVA—which is Scikit-learn's default multi-class decomposition method, too. Class-wise confidence scores for OVO can be calculated by adding the number of votes and the normalised sum of pairwise confidence levels predicted by the binary classifiers.

5.1.2 Error-Correcting Output Codes

The idea to use error-correcting output codes (ECOCs) for decomposing multi-class problems was introduced in 1995 by Dietterich and Bakiri [DB95] who took an information-theoretic perspective and framed the problem as a coding problem. To use the ECOC approach, one first defines a binary code matrix $\mathbf{W}$, in which each class is represented by one row which contains the code word for that class. Then, a classifier is trained for each column in the code matrix, but with the outcome replaced by the code matrix's corresponding entry, i.e., when fitting classifier j we replace each occurrence of class i with the entry found in row i and column j of the binary code matrix $\mathbf{W}$. To recover the n classes we apply the k classifiers, multiply the output (i.e., probability estimate) from classifier j with column vector j, and arrange the products in the same order in a matrix $\hat{\mathbf{W}}$. Finally, an observation is labelled as belonging to the class whose predicted code (i.e., row in $\hat{\mathbf{W}}$) is closest to the corresponding code (row) in the code matrix $\mathbf{W}$, according to some distance metric. The distance function proposed by Dietterich and Bakiri [DB95] is the L_1 distance

$$D(\mathbf{w}_i, \hat{\mathbf{w}}_i) = \sum_j |\hat{w}_{i,j} - w_{i,j}|, \tag{5.1}$$

where i iterates over the rows (i.e., classes) and j over the columns (i.e., binary classifiers) of the code matrix $\mathbf{W}$. A confidence score for class i can be calculated by evaluating $D(\mathbf{w}_i, \mathbf{1} - \hat{\mathbf{w}}_i)$, i.e., by calculating the L_1 distance between the code matrix and the vector of probabilities predicted by the binary classifiers for each's respective *negative* class.

Allwein, Schapire, and Singer [ASS00] subsequently extended this work—and the design space for the code matrix $\mathbf{W}$—by allowing the entries of $\mathbf{W}$ to take on one of three (instead of two) values, namely -1, 0, or $+1$, where a zero indicates that instances of this class be excluded from the corresponding model, while $+1$ and -1 encode whether the corresponding code bit is on or off, respectively. This extension makes it possible to encode any possible decomposition, including nested dichotomies, in the code matrix, but does

not provide any guidance on how to design a good code matrix. The error-correcting output codes approach has since been taken further in various papers that focus on designing a problem-dependent code matrix for a given multi-class classification problem based on training data. Pujol, Radeva, and Vitrià [PRV06], for example, proposed *Discriminant ECOC* in 2006, which uses floating search to find a nested dichotomy (binary tree) that maximises the quadratic mutual information, which is then represented as a coding matrix of size $k - 1$. More recently, Bautista et al. [Bau+12] proposed two evolutionary algorithms, based on *genetic algorithms* and *population based incremental learning*, to find a minimal coding matrix—i.e., one with $\lceil \log_2 k \rceil$ columns for a k-class problem—that achieves good generalisation for a given machine learning algorithm and classification problem.

5.1.3 Hierarchical Decomposition Strategies—Nested Dichotomies

In a nested dichotomy the k classes are placed as the k leaf nodes of a binary tree. Nested dichotomies are a well-known technique for dealing with a polychotomous response in regression analysis, whose results depend on the particular nested dichotomy used [Fox97], and which are applicable if there is enough domain knowledge to construct an appropriate and justifiable nested dichotomy for a given problem. To construct a nested dichotomy from domain knowledge (or common sense), the k classes are placed as the leaf nodes of a binary tree according to a hierarchy of the k concepts that represents the domain knowledge. To distinguish a nested dichotomy constructed from domain knowledge in this manner from one constructed by some other method, we refer to the former as an expert hierarchy and to the latter simply as a nested dichotomy. To train a nested dichotomy, an instance of the binary classifier is trained for each internal node of the tree using only the data belonging to either of the classes represented by that node's children. At prediction time, each of the trained binary classifiers is applied and the outputs aggregated. Because the dichotomies that constitute a nested dichotomy are mutually independent [Fox97], the expected probability that

a new instance belongs to a particular class is given by the product of the estimated probabilities that are on the path to the leaf representing that class.

Nested dichotomies have multiple advantages over non-hierarchical multi-class decomposition methods: lower time and space complexity at both training and evaluation (prediction) time, easier interpretation, and a modular architecture that fosters division of labour and iterative development. Time and space complexity at training time is lower for nested dichotomies than for OVA (and, by extension, than for OVO), because fewer binary classifiers need to be fitted, and because each classifier, bar the one at the root of the hierarchy, is only fitted to a subset of the training data. Time and space complexity at evaluation (prediction) time is lower for nested dichotomies, because there are fewer binary classifiers to begin with, and we may not have to evaluate all of them to predict the most likely class label.

A binary tree with k leaves has $k-1$ internal (non-leaf) nodes, and hence a nested dichotomy for a k-class problem requires fitting and storing $k-1$ binary classifiers, and evaluating between $\log_2 k$ and $k-1$ of them, depending on how often the probability predicted by an internal node's binary classifier satisfies Eq. 5.2. The number of all the possible full binary rooted trees with $n+1$ leaves is given by the n-th Catalan number [BLL98, p. 167]

$$C_n = \frac{(2n)!}{(n+1)!n!}.$$

To construct all the possible nested dichotomies for a k-class problem would thus require to fit, store, apply, and aggregate the outputs of $(k-1)C_{k-1}$ models. Because of the rapid growth of this function—for $k=4$ we have $3C_3 = 18$, for $k=7$ it is $6C_6 = 792$, and for $k=13$ we have $12C_{12} = 2496144$—considering all possible binary trees is intractable for larger values of k. Even so, if we have a sound and thorough theoretical understanding of the data generating process, or enough domain knowledge to construct a plausible nested dichotomy (or set of nested dichotomies) for a given problem, then nested dichotomies are a realistic option.

However, we often apply machine learning techniques to problems for

which we do not have enough domain knowledge to construct an appropriate nested dichotomy. To overcome this obstacle with nested dichotomies, Frank and Kramer [FK04] introduced the "Ensemble of Nested Dichotomies" in 2004, a technique that was further refined by Dong, Frank, and Kramer [DFK05] and Rodríguez, García-Osorio, and Maudes [RGM10] in 2005 and 2010, respectively. To construct an ensemble of nested dichotomies for a problem with k classes, one draws a random sample (with replacement) of predetermined size m from the space of all possible binary nested dichotomies with k leaf nodes. Each of these is then separately fitted to the data, resulting in a set of m nested dichotomies which are combined into an ensemble classifier by averaging the outputs of the individual nested dichotomies. Because an ensemble of nested dichotomies with m members is simply a combination of m nested dichotomies, it requires fitting, storing, and evaluating $m(k-1)$ fitted classifiers for a problem with k classes.

5.2 A Shortcut to Discrete Predictions for Nested Dichotomies

As we dicsussed in subsection 2.1.4, many authors use a variation of nested dichotomies that might be called a non-probabilistic nested dichotomy. The probabilistic nested dichotomies we use predict the branch probabilities at each internal node, recursively multiplying them with those predicted by its children until arriving at the leaves. A non-probabilistic nested dichotomy, on the other hand, predicts a discrete class at each internal node, only descending into the branch that corresponds to the predicted class and terminating at a single predicted activity label. Both our statistical intuition and the literature suggest that a probabilistic nested dichotomy is preferable to a non-probabilistic nested dichotomy, but non-probabilistic nested dichotomies do have one advantage. Namely, we do not need to apply all its constituent binary classifiers to predict a discrete class label, but can achieve the same outcome with $\log_2 k$ to $k-1$ classifiers, depending on whether the hierarchy is balanced or a chain, respectively. However, this aspect of probabilistic nested

dichotomies can be improved if we avoid descending into any branch whose predicted probability is too small to compete with the probabilities predicted for its sibling, or any of its sibling's descendants. This probability threshold depends on the (maximum) depth of the tree below the more likely of the two branches, and on the threshold for converting the predicted probabilities into discrete class predictions. The relationship can be formulated as follows in terms of the more likely branch's predicted probability

$$p_y \geq \frac{1}{t^d + 1}, \tag{5.2}$$

where p_y denotes the predicted probability of the more likely branch (denoted by y), t the probability threshold (assumed to satisfy $0 < t < 1$), and d the depth of the tree attached to node y, with $d = 0$ if y is itself a leaf node. We can apply Eq. 5.2 at each internal node and not descend down the less likely of its branches if the more likely branch's predicted probability p_y (which must, by definition, meet the threshold t) satisfies Eq. 5.2, in which case it is certain that the leaf with the largest predicted probability will turn out to be y if y is a leaf node, or one of y's descendants if y is an internal (classifier) node.

5.3 Computational Experiments and Evaluation

We estimate the predictive performance of five well-known multi-class decomposition methods (one-vs-all, one-vs-one, ensembles of nested dichotomies, and error-correcting output codes), five expert hierarchies, and an ensemble of expert hierarchies across five machine learning algorithms by means of stratified ten-fold cross-validation. In addition to the three algorithms which we tuned in chapter 3—gradient boosted ensembles of decision trees (GBTs), binary Support Vector Machines (SVMs), and k-Nearest Neighbours (kNN)—and logistic regression, which we employed in 4, we further broaden the scope of learning algorithms to include decision trees. We also estimate

the algorithms' performances when they are used in their multi-class formulation. Decision trees, GBTs, and kNN are all multi-class algorithms, and multinomial logistic regression a natural and well-known multi-class formulation for logistic regression. Multi-class SVMs have been formulated [WW98; CS02; JFY09], but their performance tends to be similar to that of binary SVMs with multi-class decomposition. Furthermore, fitting (and applying) a single non-linear multi-class SVM to a k-class problem tends to incur worse computational costs than fitting and applying either k binary SVMs (with one-vs-all) or $k(k-1)/2$ binary SVMs (with one-vs-one) [HL02]. We tried fitting a multi-class SVM with a polynomial kernel using the implementations due to Crammer and Singer [CS02], and to Joachims, Finley, and Yu [JFY09], but both timed out after 24 h without converging. We therefore use the linear multi-class SVM formulation from Crammer and Singer [CS02] with default hyper-parameters (C = 1.0 and $\epsilon = 1 \times 10^{-4}$).

We again standardise each feature by subtracting its mean and dividing by its standard deviation, both of which are estimated from the cross-validation fold's training data, prior to passing them to the learning algorithm. We use a random ECOC matrix with $2k = 34$ columns, which requires about twice as many classifiers as one-vs-all or an expert hierarchy, which require $k = 17$ and $k-1$ classifiers, respectively, and four times as many as one-vs-one. Five expert hierarchies, which are also used to form an ensemble of expert hierarchies, are constructed by arranging the 17 activities in the data-set as illustrated in Figures B.1–B.5. To make a fair comparison between ensembles of expert hierarchies and ensembles of nested dichotomies, we construct an ensemble of nested dichotomies with the same number of members as the ensemble of expert hierarchies, viz. five. Each expert hierarchy was constructed based on either an engineer's or an (imaginary) user's intuition. The engineer's intuition is to split classes such that the splits are easy to learn for an algorithm, for example because they result in similar patterns in the data. This perspective is represented by EH1 and EH3 (Figures B.1 and B.3). A user's intuition, on the other hand, is to split classes such that the earlier splits, which are higher up in the hierarchy, are more informative to them than later splits that are further towards the hierarchy's leaves. This perspective is represented by

EH2 (Figure B.2), which considers fall detection, EH4 (Figure B.4), which considers separating potential emergencies (in an emergency first response context) from normal behaviours, and EH5 (Figure B.5), which considers detecting when someone ascends or descends the stairs.

We further compare multi-class decomposition method (and multi-class) performance on the two binary classification problems corresponding to the topmost (root) dichotomy of EH1 (an example of an engineer's expert hierarchy) and EH4 (an example of a user's expert hierarchy). The former dichotomy separates "Stationary" from "Mobile," and the latter "Possible Emergency" from "Not Emergency" activities. Incidentally, these two splits also provide examples of different levels of class imbalance, with the EH1 split leading to a moderately imbalanced (67%/33%) and the EH4 split to a heavily imbalanced (89%/11%) data-set. The confidence scores obtained with multi-class decomposition methods based on nested dichotomies such as ensembles of nested dichotomies, expert hierarchies, and ensembles of expert hierarchies are true multi-class probabilities (as far as the binary classifiers are able to estimate their binary probabilities), and the confidence scores obtained with one-vs-all can easily be combined into multi-class probabilities, but the confidence scores estimated by one-vs-one and error-correcting output codes do not share this characteristic and are prone to be severely affected by class imbalance. To overcome this issue, and give these multi-class decomposition methods a chance to compete on the EH1 and EH4 dichotomies, we calibrate their scores—as well as those estimated by SVM, which is not designed to estimate probabilities even in the binary case—via Platt scaling [Pla99].

The computational experiments were implemented in Python (version 3.7.3), using the sklearn [Ped+11, version 0.20] implementations of machine learning algorithms and multi-class decomposition methods where available (i.e., one-vs-one, one-vs-all, error-correcting output codes, and all machine learning algorithms), and writing our own where necessary, namely for the expert hierarchies, ensembles of expert hierarchies, and ensembles of nested dichotomies. To speed up the experiments they were parallelised using GNU Parallel [Tan11].

5.4 Results

This section presents and analyses the results of the experiments described in section 5.3. We use Cohen's Kappa (κ) statistic as our metric of predictive performance because of its inherent ability to quantify a classifier's performance on a multi-class classification problem, and because it is adjusted for the prior class distributions of both the ground truth and the predicted class labels.

For a detailed analysis of the differences between the various combinations of machine learning algorithms and multi-class decomposition methods we employ (binomial) logistic regression of the κ statistic on the two factors of interest, viz. the learning algorithm and multi-class decomposition method. The κ statistic, calculated once for each cross-validation test fold, corresponds to the proportion of successful Bernoulli trials—the proportion of test instances classified correctly, adjusted for the probability of chance agreement—and the number of instances in a test fold to the number of trials. Together, these two numbers determine the binomial distribution, allowing us to apply a (binomial) logistic regression model to estimate the log-odds of the κ statistic, $\eta = \ln \frac{\kappa}{1-\kappa}$, which relate to the κ statistic via the logistic function

$$\kappa = g(\eta) = \frac{e^{\eta}}{e^{\eta} + 1}.$$

Because OVA is by far the most popular multi-class decomposition method in practice, and GBT the algorithm that is most likely to outperform the others, we use that combination (OVA with GBT) as the baseline (i.e., the regression equation's intercept) against which the other combinations of multi-class decomposition methods and algorithms are compared. The models were fitted using the R Language and Environment for Statistical Computing [RCT19, version 3.6.1]. In our analysis we limit ourselves to those regression coefficients that are significant at the $\alpha = 0.1$ significance level.

Table 5.1 shows the mean κ (in percent, $\pm$ its SE) across the ten cross-validation folds for each multi-class decomposition method. The column labelled "Avg." lists the mean and SE for each multi-class decomposition

Table 5.1: Mean κ% ($\pm$ SE) on the multi-class HAR problem for each machine learning algorithm and multi-class decomposition method. The first six multi-class decomposition methods are given in order of decreasing average score, the expert hierarchies ordered alphabetically.

	GBT	SVM	DT	kNN	GLM	SVM-MCL	Avg.
OVO	95.37 ± 0.29	90.88 ± 0.26	90.56 ± 0.20	83.37 ± 0.24	87.35 ± 0.47	87.10 ± 0.36	89.11 ± 1.68
END	95.30 ± 0.27	91.08 ± 0.31	92.19 ± 0.33	85.88 ± 0.34	81.37 ± 0.47	82.34 ± 0.51	88.03 ± 2.32
OVA	95.24 ± 0.25	90.67 ± 0.33	89.41 ± 0.45	85.45 ± 0.24	83.43 ± 0.44	83.88 ± 0.53	88.01 ± 1.88
EEH	95.31 ± 0.22	90.29 ± 0.34	87.06 ± 0.34	85.43 ± 0.32	84.89 ± 0.32	84.85 ± 0.33	87.97 ± 1.69
MCL	95.85 ± 0.20	-	84.21 ± 0.49	85.45 ± 0.24	85.67 ± 0.50	80.72 ± 0.47	86.38 ± 2.53
ECOC	93.91 ± 0.21	90.33 ± 0.32	95.84 ± 0.20	85.74 ± 0.27	73.93 ± 0.29	71.56 ± 0.56	85.22 ± 4.20
Avg.	95.16 ± 0.27	90.65 ± 0.15	89.88 ± 1.65	85.22 ± 0.38	82.77 ± 1.95	81.74 ± 2.22	87.45 ± 0.57
EH1	94.95 ± 0.21	89.42 ± 0.39	85.29 ± 0.19	85.36 ± 0.33	83.37 ± 0.34	83.51 ± 0.30	86.98 ± 1.83
EH2	94.87 ± 0.24	89.64 ± 0.30	85.29 ± 0.39	85.54 ± 0.28	83.51 ± 0.37	83.40 ± 0.34	87.04 ± 1.82
EH3	94.81 ± 0.21	89.94 ± 0.41	85.68 ± 0.35	85.55 ± 0.26	83.62 ± 0.37	83.65 ± 0.30	87.21 ± 1.79
EH4	94.76 ± 0.24	89.84 ± 0.38	85.45 ± 0.28	85.43 ± 0.30	82.99 ± 0.50	83.10 ± 0.39	86.93 ± 1.87
EH5	94.69 ± 0.31	89.65 ± 0.25	84.93 ± 0.32	85.39 ± 0.25	81.11 ± 0.37	80.89 ± 0.36	86.11 ± 2.16
Avg.	94.82 ± 0.04	89.70 ± 0.09	85.33 ± 0.12	85.45 ± 0.04	82.92 ± 0.46	82.91 ± 0.51	86.85 ± 0.19

Table 5.2: Mean κ% ($\pm$ SE) for the topmost dichotomy of EH1 (Stationary vs. Mobile). The first six multi-class decomposition methods are given in order of decreasing average score, the expert hierarchies in alphabetical order.

	GBT	SVM	DT	kNN	GLM	SVM-MCL	Avg.
EEH	99.85 ± 0.06	99.77 ± 0.09	99.67 ± 0.08	99.14 ± 0.16	99.72 ± 0.10	99.70 ± 0.07	99.64 ± 0.10
OVO	99.85 ± 0.06	99.70 ± 0.07	99.80 ± 0.06	99.19 ± 0.12	99.52 ± 0.10	99.62 ± 0.08	99.61 ± 0.10
END	99.77 ± 0.09	99.65 ± 0.10	99.59 ± 0.08	99.11 ± 0.15	99.14 ± 0.11	99.06 ± 0.13	99.39 ± 0.13
MCL	99.75 ± 0.08	-	99.32 ± 0.11	99.09 ± 0.17	99.52 ± 0.14	98.43 ± 0.25	99.22 ± 0.23
ECOC	99.77 ± 0.07	99.42 ± 0.11	99.80 ± 0.06	98.99 ± 0.16	96.11 ± 0.31	93.87 ± 0.87	97.99 ± 1.00
OVA	99.80 ± 0.08	99.27 ± 0.20	93.02 ± 0.44	99.09 ± 0.17	98.78 ± 0.13	97.80 ± 0.38	97.96 ± 1.02
Avg.	99.80 ± 0.02	99.56 ± 0.09	98.53 ± 1.11	99.10 ± 0.03	98.80 ± 0.55	98.08 ± 0.89	98.97 ± 0.32
EH1	99.82 ± 0.07	99.72 ± 0.10	99.39 ± 0.09	99.09 ± 0.17	99.60 ± 0.08	99.57 ± 0.07	99.53 ± 0.11
EH2	99.77 ± 0.06	99.39 ± 0.17	98.66 ± 0.14	99.06 ± 0.15	98.78 ± 0.15	98.66 ± 0.15	99.05 ± 0.18
EH3	99.82 ± 0.07	99.70 ± 0.08	99.52 ± 0.10	99.06 ± 0.17	99.62 ± 0.10	99.47 ± 0.12	99.53 ± 0.11
EH4	99.82 ± 0.07	99.72 ± 0.10	99.26 ± 0.16	99.11 ± 0.16	99.49 ± 0.12	99.49 ± 0.12	99.48 ± 0.11
EH5	99.80 ± 0.07	99.52 ± 0.12	99.14 ± 0.15	99.09 ± 0.17	99.06 ± 0.13	98.91 ± 0.16	99.25 ± 0.14
Avg.	99.81 ± 0.01	99.61 ± 0.07	99.19 ± 0.15	99.08 ± 0.01	99.31 ± 0.17	99.22 ± 0.18	99.37 ± 0.09

Table 5.3: Mean κ% ($\pm$ SE) for the topmost dichotomy of EH4 (Possible Emergency vs. non-Emergency). The first six multi-class decomposition methods are given in order of decreasing average score, the five expert hierarchies in alphabetical order.

	GBT	SVM	DT	kNN	GLM	SVM-MCL	Avg.
EEH	94.11 ± 0.61	92.46 ± 0.47	87.99 ± 0.66	86.86 ± 0.79	86.71 ± 0.82	87.59 ± 0.70	89.29 ± 1.30
OVO	94.93 ± 0.48	90.90 ± 0.73	89.69 ± 0.63	82.41 ± 1.02	89.32 ± 0.87	88.41 ± 0.93	89.28 ± 1.66
END	94.34 ± 0.51	92.68 ± 0.54	90.66 ± 0.61	86.97 ± 0.89	83.44 ± 1.57	84.79 ± 1.30	88.81 ± 1.80
MCL	94.74 ± 0.57	-	81.38 ± 1.22	87.35 ± 0.72	88.18 ± 0.57	80.87 ± 0.86	86.50 ± 2.54
ECOC	94.83 ± 0.30	91.19 ± 0.57	92.69 ± 0.39	85.65 ± 0.59	77.26 ± 0.90	76.34 ± 1.41	86.33 ± 3.26
OVA	94.36 ± 0.50	92.91 ± 0.56	59.74 ± 1.34	87.35 ± 0.72	86.20 ± 1.12	84.89 ± 1.05	84.24 ± 5.14
Avg.	94.55 ± 0.13	92.03 ± 0.41	83.69 ± 5.04	86.10 ± 0.78	85.19 ± 1.78	83.81 ± 1.84	87.41 ± 0.84
EH1	94.78 ± 0.51	91.30 ± 0.69	82.30 ± 0.61	86.95 ± 0.67	84.35 ± 1.24	84.39 ± 1.08	87.34 ± 1.95
EH2	94.06 ± 0.54	91.00 ± 0.55	84.02 ± 0.70	87.70 ± 0.67	84.62 ± 0.83	83.93 ± 0.88	87.55 ± 1.72
EH3	93.16 ± 0.68	92.39 ± 0.61	83.54 ± 0.79	87.54 ± 0.74	85.52 ± 0.84	86.17 ± 0.69	88.05 ± 1.59
EH4	91.75 ± 0.69	90.53 ± 0.44	81.74 ± 0.59	87.35 ± 0.72	79.98 ± 1.36	81.15 ± 1.23	85.42 ± 2.09
EH5	93.65 ± 0.58	90.82 ± 0.55	81.81 ± 0.94	87.48 ± 0.84	82.43 ± 1.03	82.97 ± 0.83	86.53 ± 2.01
Avg.	93.48 ± 0.51	91.21 ± 0.32	82.68 ± 0.46	87.40 ± 0.13	83.38 ± 0.99	83.72 ± 0.83	86.98 ± 0.46

method, computed across the five machine learning algorithms, and the two rows labelled "Avg." the mean and SE over the preceding five rows. Figure 5.1 illustrates normal (Gaussian) 99% confidence intervals (C.I.s) calculated from the means and standard errors given in Table 5.1. Clearly, the variance between expert hierarchies is negligible compared to that between the other multi-class decomposition methods, and there is no a priori reason to prefer any particular expert hierarchy over the others. Therefore, we pooled the five expert hierarchies (EH1, EH2, ..., EH5) into a single category labelled "EH," and then fitted the regression model to the data summarised in Table 5.1 to estimate coefficients for seven, rather than eleven, multi-class decomposition methods—OVA (the baseline/intercept), OVO, ensembles of nested dichotomies (ENDs), ECOC, multi-class (MCL), expert hierarchies (EHs) with no distinction between individual hierarchies), and ensembles of expert hierarchies (EEH)—and five machine learning algorithms, namely ensembles of gradient boosted trees (the baseline/intercept), (binary) SVMs, multi-class SVMs (SVM-MCL), decision trees (DTs), kNN, and logistic regression (GLM).

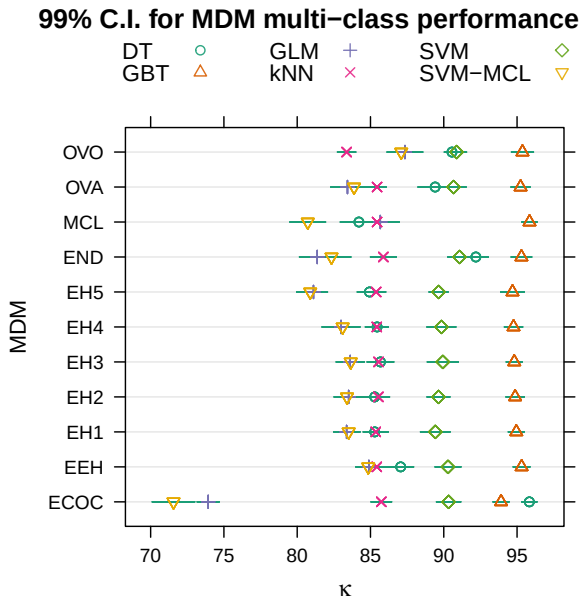


Figure 5.1: 99% C.I.s for the effect of the multi-class decomposition method (MDM) on the Kappa statistic for the full 17-class problem

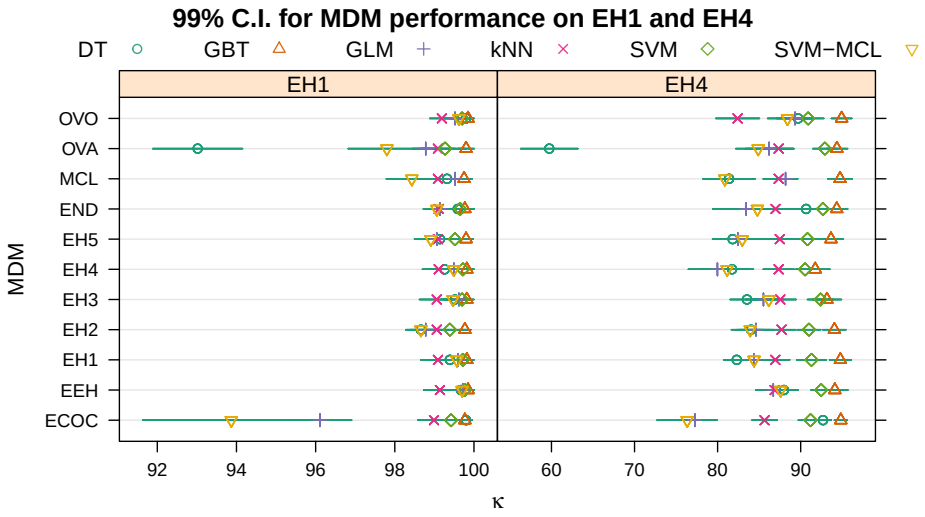


Figure 5.2: 99% C.I.s for the effect of the multi-class decomposition method (MDM) on the Kappa statistic for the topmost dichotomy of EH1 (left) and EH4 (right)

Tables 5.2 and 5.3 show the mean κ ($\pm$ SE), in percent, when evaluating each multi-class decomposition method/machine learning algorithm combination on the topmost (root) dichotomy of EH1 ("Stationary" vs. "Mobile") and EH4 ("Possible Emergency" vs. "Not Emergency"), respectively. The column labelled "Avg." lists the mean κ ($\pm$ SE), again in percent, across the five machine learning algorithms for each multi-class decomposition method, and the rows labelled "Avg." the mean and its SE across the preceding five rows. Figure 5.2 illustrates the 99% C.I.s for each combination of multi-class decomposition method and learning algorithm based on the means and standard errors in Tables 5.2 and 5.3.

The results of our analysis of the data from the multi-class problem summarised in Table 5.1 are given in Table 5.4, and those for the dichotomous problems induced by EH1 and EH4 (Table 5.2 and 5.3) are given in Table 5.5 and 5.6, respectively. The tables list the estimate (β) along with its 99% C.I. and P-value for those coefficients that are significant at the $\alpha = 0.1$ level, i.e., those with $P < 0.1$. The row labelled "(Intercept)" corresponds to the baseline method's (OVA $\wedge$ GBT) estimated log odds. For example, the log odds for

OVA ∧ GBT on the multi-class problem are estimated as $\beta \approx 2.99$. Therefore, the odds ratio is $e^{\beta} \approx e^{2.99} \approx 19.9$ and hence $\kappa \approx 100\frac{19.9}{19.9+1} \approx 95.2\%$. The other coefficients' estimates and C.I.s indicate the marginal change in log-odds associated with the corresponding multi-class decomposition method (MDM), learning algorithm, or combination of multi-class decomposition method and learning algorithm. Note that because a positive coefficient signifies an increase and a negative coefficient a decrease in the odds, coefficients whose C.I.s span zero are not significant at the $\alpha = 0.01$ significance level. Coefficients labelled with a multi-class decomposition method, rather than a combination of multi-class decomposition method and algorithm, estimate the marginal effect that the multi-class decomposition method has on algorithm performance and therefore apply when the multi-class decomposition method is combined with any of the algorithms. Conversely, coefficients labelled with an algorithm, rather than a combination of algorithm and multi-class decomposition method, estimate the marginal effect that the algorithm has on multi-class decomposition method performance, and thus apply when the algorithm is combined with any of the multi-class decomposition methods. Finally, these independent multi-class decomposition method and algorithm coefficients may be amplified or attenuated by a coefficient labelled with a combination of multi-class decomposition method and algorithm ("MDM ∧ algorithm"). These interaction coefficients apply in addition to the independent multi-class decomposition method and algorithm coefficients.

The following examples serve to illustrate these concepts. Consider the logistic regression (GLM) estimates for the multi-class problem from Table 5.4. The "(Intercept)" (GBT ∧ OVA) is estimated at 2.99, corresponding to odds of $e^{2.99} \approx 19.9$, and hence to a mean κ of $e^{2.99}/(e^{2.99}+1) \approx 19.9/(19.9+1) \approx 95.2\%$. An estimate of -1.38 means that the GLM odds are $e^{-1.38} \approx 0.25$ times the baseline odds, i.e., $e^{-1.38}e^{2.99} = e^{2.99-1.38} \approx 5.0$ which is equivalent to a mean κ of $e^{2.99-1.38}/(e^{2.99-1.38}+1) \approx 83.3\%$. This estimate does not significantly change when GLM is combined with an expert hierarchy or an ensemble of expert hierarchies, as is attested by the absence of the corresponding coefficients from Table 5.4. However, when GLM is combined with an END, the estimated odds change by a factor of $e^{-0.15} \approx 0.861$,

corresponding to a change of $100(0.861) - 100 = -13.9\%$ and a mean κ of $e^{2.99-1.38-0.15}/(e^{2.99-1.38-0.15}+1) \approx 81.2\%$. Note that DT is the only other algorithm whose END performance is significantly different (by a factor of $e^{0.32} \approx 1.377$) from its baseline (one-vs-all) performance. When GLM is applied in its multi-class formulation its odds are subject to the multi-class effect (MCL) that applies to all algorithms, estimated as a $100e^{0.15} - 100 \approx 16.2\%$ change, which corresponds to a mean κ of $e^{2.99+0.15-1.38}/(e^{2.99+0.15-1.38}+1) \approx 85.3\%$ for logistic regression. Note that an estimate of -0.15 for the "MCL $\wedge$ kNN" coefficient means that the 16.2% improvement does not hold for kNN, and that an estimate of -0.61 for the "MCL $\wedge$ DT" coefficient, which equates to a $e^{0.15-0.61} \approx -36.9\%$ change in the odds, means that decision trees perform better with one-vs-all than in their multi-class formulation. Finally, let us consider the "ECOC $\wedge$ GLM" combination. When combined with one-vs-all, the log-odds for GLM are $2.99 - 1.38 \approx 1.61$. This baseline estimate is subject to the -0.26 change associated with error-correcting output codes overall, and an additional -0.31 change specific to the "ECOC $\wedge$ GLM" interaction, accumulating in odds that are only $100e^{-0.26-0.31} = e^{-0.57} \approx 56.6\%$, equivalent to a mean κ of $100e^{1.61-0.57}/(e^{1.61-0.57}+1) \approx 73.9\%$, of logistic regression's baseline odds.

5.5 Discussion

Our analysis shows that the ensemble of gradient boosted trees significantly and consistently outperforms the other algorithms, both on the original 17-class problem and on the two dichotomous problems induced by the topmost dichotomy of EH1 and EH4. On all three problems, the next best learning algorithm tends to be SVM, followed by decision trees, kNN, and finally logistic regression and the multi-class SVM. While there is no such clear ranking for the multi-class decomposition methods, there are some discernible patterns.

Logistic regression, decision trees, and the multi-class SVM are more sensitive to the choice of multi-class decomposition method than the other learning algorithms. Decision trees consistently achieve their best performance

Table 5.4: Estimated logistic regression coefficients with $P < 0.1$ for the multi-class problem

Coefficient	0.5%	β	99.5%	P
(Intercept)	2.87	2.99	3.13	$<2.0 \times 10^{-32}$
SVM	−0.88	−0.72	−0.56	1.0×10^{-31}
DT	−1.02	−0.86	−0.70	$<2.0 \times 10^{-32}$
kNN	−1.38	−1.22	−1.08	$<2.0 \times 10^{-32}$
GLM	−1.53	−1.38	−1.23	$<2.0 \times 10^{-32}$
SVM-MCL	−1.50	−1.35	−1.20	$<2.0 \times 10^{-32}$
ECOC	−0.43	−0.26	−0.09	9.6×10^{-5}
ECOC ∧ SVM	0.00	0.22	0.44	8.6×10^{-3}
ECOC ∧ DT	1.03	1.26	1.50	$<2.0 \times 10^{-32}$
ECOC ∧ kNN	0.08	0.28	0.49	3.5×10^{-4}
ECOC ∧ GLM	−0.51	−0.31	−0.12	3.8×10^{-5}
ECOC ∧ SVM-MCL	−0.66	−0.47	−0.27	9.3×10^{-10}
EEH ∧ DT	−0.46	−0.24	−0.02	4.3×10^{-3}
EH ∧ DT	−0.46	−0.28	−0.11	1.9×10^{-5}
END ∧ DT	0.09	0.32	0.55	3.0×10^{-4}
END ∧ GLM	−0.36	−0.15	0.05	5.5×10^{-2}
MCL	−0.04	0.15	0.33	4.6×10^{-2}
MCL ∧ DT	−0.83	−0.61	−0.38	2.4×10^{-12}
MCL ∧ kNN	−0.36	−0.15	0.07	8.5×10^{-2}
OVO ∧ kNN	−0.40	−0.19	0.02	2.2×10^{-2}
OVO ∧ GLM	0.07	0.29	0.50	5.4×10^{-4}
OVO ∧ SVM-MCL	0.02	0.23	0.44	5.2×10^{-3}

Table 5.5: Estimated logistic regression coefficients with $P < 0.1$ for the binary problem induced by the topmost dichotomy of EH1

Coefficient	0.5%	β	99.5%	P
(Intercept)	5.65	6.20	6.87	$<2.0 \times 10^{-32}$
SVM	−2.03	−1.29	−0.64	1.2×10^{-6}
DT	−4.29	−3.61	−3.05	$<2.0 \times 10^{-32}$
kNN	−2.24	−1.51	−0.88	6.2×10^{-9}
GLM	−2.52	−1.80	−1.19	1.3×10^{-12}
SVM-MCL	−3.10	−2.40	−1.82	1.6×10^{-22}
ECOC ∧ DT	2.71	3.73	4.80	1.9×10^{-20}
ECOC ∧ GLM	−1.95	−1.07	−0.17	1.8×10^{-3}
ECOC ∧ SVM-MCL	−1.81	−0.95	−0.07	4.6×10^{-3}
EEH ∧ SVM	−0.26	0.89	2.03	4.4×10^{-2}
EEH ∧ DT	1.77	2.83	3.88	3.0×10^{-12}
EEH ∧ GLM	0.09	1.20	2.29	4.7×10^{-3}
EEH ∧ SVM-MCL	0.62	1.71	2.78	3.6×10^{-5}
EH ∧ SVM	−0.15	0.58	1.39	4.9×10^{-2}
EH ∧ DT	1.52	2.17	2.91	4.7×10^{-16}
EH ∧ GLM	−0.17	0.52	1.30	6.4×10^{-2}
EH ∧ SVM-MCL	0.33	1.00	1.75	2.7×10^{-4}
END ∧ SVM	−0.15	0.85	1.87	2.9×10^{-2}
END ∧ DT	2.09	3.03	3.99	1.3×10^{-16}
END ∧ SVM-MCL	0.08	0.99	1.90	4.8×10^{-3}
MCL ∧ DT	1.73	2.61	3.52	2.8×10^{-14}
MCL ∧ GLM	0.23	1.16	2.12	1.4×10^{-3}
OVO ∧ DT	2.20	3.32	4.45	1.4×10^{-14}
OVO ∧ SVM-MCL	0.42	1.49	2.53	2.4×10^{-4}

Table 5.6: Estimated logistic regression coefficients with $P < 0.1$ for the binary problem induced by the topmost dichotomy of EH4

Coefficient	0.5%	β	99.5%	P
(Intercept)	2.70	2.82	2.94	$<2.0 \times 10^{-32}$
SVM	−0.40	−0.25	−0.09	7.0×10^{-5}
DT	−2.56	−2.42	−2.29	8.9×10^{-2}
kNN	−1.03	−0.88	−0.74	$<2.0 \times 10^{-32}$
GLM	−1.13	−0.99	−0.84	$<2.0 \times 10^{-32}$
SVM-MCL	−1.23	−1.09	−0.95	$<2.0 \times 10^{-32}$
ECOC ∧ SVM	−0.55	−0.33	−0.10	1.6×10^{-4}
ECOC ∧ DT	1.85	2.05	2.26	$<2.0 \times 10^{-32}$
ECOC ∧ kNN	−0.44	−0.24	−0.03	2.9×10^{-3}
ECOC ∧ GLM	−0.90	−0.70	−0.50	1.3×10^{-19}
ECOC ∧ SVM-MCL	−0.84	−0.65	−0.45	4.1×10^{-17}
EEH ∧ DT	1.45	1.64	1.84	$<2.0 \times 10^{-32}$
EEH ∧ SVM-MCL	0.07	0.27	0.47	4.3×10^{-4}
EH	−0.28	−0.15	−0.03	2.0×10^{-3}
EH ∧ DT	1.18	1.32	1.47	$<2.0 \times 10^{-32}$
EH ∧ kNN	0.00	0.16	0.32	9.0×10^{-3}
END ∧ DT	1.68	1.88	2.08	$<2.0 \times 10^{-32}$
END ∧ GLM	−0.41	−0.21	−0.01	6.0×10^{-3}
MCL ∧ DT	0.81	1.01	1.20	$<2.0 \times 10^{-32}$
OVO	−0.06	0.11	0.29	8.9×10^{-2}
OVO ∧ SVM	−0.61	−0.38	−0.16	8.9×10^{-6}
OVO ∧ DT	1.45	1.66	1.86	$<2.0 \times 10^{-32}$
OVO ∧ kNN	−0.70	−0.50	−0.30	2.2×10^{-10}
OVO ∧ GLM	−0.03	0.18	0.39	2.7×10^{-2}
OVO ∧ SVM-MCL	−0.01	0.19	0.40	1.6×10^{-2}

when combined with error-correcting output codes. In fact, combining decision trees with error-correcting output codes achieves a κ on the 17-class problem that is only 0.01 percentage points lower than the 95.82% achieved by a multi-class ensemble of gradient boosted trees, our best result on this problem. With any other algorithm, error-correcting output codes perform comparably or worse than one-vs-all, making it one of the worse multi-class decomposition methods for this problem. This is particularly true for logistic regression, which achieves its worst result on all three problems with error-correcting output codes.

One-vs-one, which many studies found to perform slightly better than one-vs-all, does not consistently outperform one-vs-all in our evaluation, nor does it achieve the top result for any of our three classification problems. One-vs-one performs significantly (at the $\alpha = 0.01$ significance level) better than one-vs-all on the 17-class problem when combined with logistic regression or the multi-class SVM, the EH1 dichotomy when combined with decision trees or the multi-class SVM, and on the EH4 dichotomy when combined with decision trees. Furthermore, one-vs-one achieves significantly worse performance on the EH4 problem when combined with SVM, where it achieves 31.6% lower odds than one-vs-all, or kNN, where it achieves 39.3% lower odds than one-vs-all. Applying an algorithm's multi-class formulation performs significantly (at the $\alpha = 0.01$ significance level) better than one-vs-all on the topmost EH1 dichotomy when combined with decision trees or logistic regression, and on the topmost EH4 dichotomy when combined with decision trees. Otherwise, an algorithm's multi-class formulation performs comparably to one-vs-all.

Performance varies much less among expert hierarchies than among the other multi-class decomposition methods, which indicates that any reasonable expert hierarchy is a reasonable choice, and searching for better hierarchies is unlikely to yield significant improvements. Expert hierarchies perform comparably or better than one-vs-all with most algorithms on all three problems. One exception is decision trees, which achieve 24.4% lower odds on the 17-class problem with expert hierarchies than with one-vs-all. The other exceptions are SVM (both in its binary and multi-class formulation), the ensemble of gradient boosted trees, and logistic regression, all of which achieve

13.9% lower odds on the topmost dichotomy of EH4 with expert hierarchies than with one-vs-all. Ensembles of nested dichotomies perform comparably or better than one-vs-all with all but one algorithm. That exception is logistic regression on both the 17-class problem, where it achieves 13.9% lower odds with an ensemble of nested dichotomies than with one-vs-all, and the binary problem induced by the topmost dichotomy of EH4, where it achieves 18.9% lower odds than with one-vs-all. Ensembles of expert hierarchies, on the other hand, perform comparably or better than one-vs-all with all algorithms on all three problems. This makes an ensemble of expert hierarchies a better multi-class decomposition method for this problem than an arbitrary ensemble of (random) nested dichotomies, which may be more difficult to justify to a domain expert.

These results show that expert hierarchies can compete with other multi-class decomposition methods and inherent multi-class classifiers. As we have mentioned, expert hierarchies have two main advantages over both multi-class classifiers and domain-agnostic multi-class decomposition methods. The first advantage is iterative and modular development, and the second is targeted tuning and optimisation.

Iterative and modular development can speed up and facilitate many of the tasks involved in designing, developing, and maintaining and improving a HAR system. One of the most time consuming tasks when developing a HAR system is often data annotation. With an inherent multi-class classification algorithm, predictive modelling must wait until a data-set has been annotated with *all* the activities of interest, and be repeated if a new activity is introduced. New activities can be introduced if a requirement emerges to distinguish between different types of some higher-level activity.

For example, it might be decided upon further consultation with professionals that a HAR system developed for monitoring firefighters' during operations or training really ought to distinguish between crawling on one's hands and knees, and military style on one's stomach. This distinction may well be an important one, because smoke tends to rise which makes it important to keep as close to the ground as possible.

With one-vs-all it is possible, at least in principle, to begin modelling as

soon as the annotations for one class (say, standing) are complete. However, the class imbalance inherent to a one-vs-all decomposition (e.g., "standing" vs. "not standing") means that any insights gleaned from the modelling will be heavily biased and may not apply to the other dichotomisers. Furthermore, it is probably less efficient and possibly more error-prone to go through a data-set (e.g., fast-forward through hours of video footage) and annotate every time the subject is, or ceases to be, standing, than to annotate when subjects transition between, for example, stationary and mobile behaviour.

With expert hierarchies, annotators can generate high-level annotations (e.g., stationary versus mobile) and hand them over to the data science team. The data scientists can then develop and tune the top-level discriminator, knowing that the degree to which they succeed in developing an accurate discriminator for the given labels is directly linked to the system's overall accuracy. Furthermore, the independence of the dichotomisers that constitute an expert hierarchy makes it possible to replace any of them with a pre-trained model. This means that it is in principle possible to integrate models that have been developed by a third party and fitted to data private or confidential to them, be it to improve the expert hierarchy by replacing an existing dichotomiser or to extend the expert hierarchy with the capacity to make a finer-grained distinction by replacing a leaf in the expert hierarchy with a specialised dichotomiser.

Targeted tuning and optimisation of HAR inference capabilities makes it possible to not only identify problematic activities (e.g., activities with high misclassification costs that tend to be confused with each other), but to effectively improve the performance on the problematic activities without negatively affecting performance on the other activities. Each dichotomiser in an expert hierarchy is an independent binary classifier whose performance can not only be analysed and tuned, but which can be swapped out for a different algorithm. If the resulting dichotomiser is more accurate than the one it is replacing, then it is bound to improve the multi-class performance. While it is easy to aggregate the probabilities predicted by a true multi-class classifier or some multi-class decomposition method according to an expert hierarchy, we cannot map performance at some internal node of the hierarchy to a

single classifier. The independence between an expert hierarchy's constituent dichotomies makes it easier to explain a prediction to someone without a background in machine learning. Instead of having to simultaneously examine and balance the predicted probabilities of multiple classifiers, none of which says much about the probability distribution over all classes, we can easily identify and examine the output of the binary classifier corresponding to the level at which the prediction first went wrong. Because that classifier is independent of its ancestors and because its own performance has no effect on its descendants', we can focus our efforts on improving a single binary classifier without having to worry about negatively affecting the performance on other classes.

5.6 Conclusions

In this chapter, we presented the first empirical comparison of different multi-class decomposition methods, as well as inherently multi-class classifiers, for human activity recognition, which covers not only the most popular methods from the literature, namely one-vs-all, one-vs-one, error-correcting output codes, and ensembles of nested dichotomies, but also nested dichotomies that are constructed from domain knowledge, which we call expert hierarchies, and ensembles of expert hierarchies. An expert hierarchy has the advantage that it requires one less binary classifier than one-vs-all, which requires k classifiers to represent a k-class problem, and that it results in a multi-class decomposition that is easier to interpret than that resulting from one-vs-all. In particular, an expert hierarchy can be designed such that it separates the two most important general concepts—for example "Potential Emergency" and "Not An Emergency"—first, i.e., at the topmost level of the hierarchy. With an expert hierarchy it is possible to obtain an estimate for the topmost dichotomy by applying only one model (the one corresponding to the topmost dichotomy), which is impossible with any other multi-class decomposition method. We demonstrated this scenario by comparing the predictive performance on the binary classification problem induced by the topmost dichotomy of two example expert hierarchies. Finally, we formulated a threshold that can be

used to further reduce the computational complexity of predicting the most likely class label with expert hierarchies—or any nested dichotomy, since an expert hierarchy is just a special case of a nested dichotomy.

Our results show that expert hierarchies perform comparably to one-vs-all, both on the original multi-class problem and on more general binary classification problems such as those induced by expert hierarchies' topmost dichotomy. Our results further show that individual expert hierarchies tend to perform similarly, particularly when compared to the much larger variance between other multi-class decomposition methods or between learning algorithms. When multiple expert hierarchies are combined into an ensemble, they perform comparably to one-vs-one and better than one-vs-all on the full multi-class problem, and outperform all multi-class decomposition methods on the two dichotomous problems. Because an expert hierarchy's constituent dichotomisers are independent of each other it is possible to analyse and optimise each dichotomiser in isolation. This enables modular and iterative development of increasingly complex HAR capabilities, which is a pre-requisite for agile development techniques, and for targeted tuning and optimisation of the resulting HAR system.

Chapter 6

Detecting Human Movement from Ambient Wi-Fi Signal Strength with Micro Models

Detecting the presence of human occupants is a critical task in building-automation systems and their interfaces, including, for example, efficient heating & ventilation, responsive lighting, home security, or ambient assisted living. For convenience and generality, their operation should not require occupants to carry or wear devices (e.g., smart phones, active badges, or Bluetooth-enabled wearables), which can be inconvenient, are easily lost or forgotten, and run out of battery. Device-free sensing circumvents these issues by abandoning user-worn devices altogether, and instead exploits signals from environmental sensors. A particularly valuable environmental sensing modality for device-free human sensing are radio signals, such as those used by Wi-Fi networks, which are of special interest due to their widespread use in both residential and commercial settings.

When we reviewed the literature on device-free human presence detection in section 2.2, we saw that most of the proposed methods, one way or another, rely on data from the Wi-Fi links in the target environment. Some, such as RASID [KSY12], learn from a small data sample from the vacant area of interest to detect deviant patterns, which are labelled as human presence.

Others, such as the one proposed by Zhou, Yang, Wu, Shangguan, and Liu [Zho+14], require samples not only of the vacant area of interest, but also of human presence at various clearly defined locations around each Wi-Fi receiver. Even unsupervised methods such as FIMD [Xia+12], which do not rely on any *labelled* data, require a substantial and representative data-set from the involved wireless links in their target environment to form accurate clusters or find reliable patterns. The few papers that do propose methods that are designed to be trained in one environment and deployed anywhere else evaluate their approach only in a few rooms with comparable dimensions, and without consideration for other changes in the environment, such as moved furniture or Wi-Fi transceivers. Finally, the Wi-Fi transceivers' placement—or at least knowledge of their relative locations—is an integral, but often implicit, assumption of most of the proposed methods. To deploy such a system one has to survey each target environment, decide where to place the transceivers, physically deploy the transceivers, and—of course—maintain them throughout the system's lifetime. In its reliance on dedicated Wi-Fi nodes, these approaches fail to exploit the ubiquity of Wi-Fi networks, and are at odds with the real-world requirements of deploying device-free sensing systems at scale.

In this chapter, we develop methods that can reliably detect the presence of a moving human occupant in a room using low cost off-the-shelf components. We make no assumptions about the Wi-Fi network's topology or geometry, beyond there being a Wi-Fi access point (AP) with multiple antennae and at least one transmitting client in the monitored room. By opportunistically taking advantage of whatever clients are transmitting to the AP, we significantly improve human presence detection accuracy over using an arbitrary single AP-client link. We demonstrate that it is possible to compete with methods that are based on channel state information (CSI) data, but only using the antenna-wise received signal strength (RSS) captured by an AP serving a regular Wi-Fi network, which circumvents the elaborate preprocessing and denoising required with CSI signals. The antenna-wise RSS differs from the total RSS, which the (IEEE 802.11) Wi-Fi standards codify as received signal strength indicator (RSSI), in that the former measures the RSS

in units such as decibel or microvolt at each antenna individually, whereas the latter measures the sum of the RSS across all the antennae in arbitrary device-dependent units. These differences in how RSSI is measured have been shown to be significant (in the context of device-free human sensing) not only among different chipset manufacturers and models, but even among chipsets of the same model [Lui+11]. We show that one can train a model to detect human movement in one room, using a carefully selected link's antenna-wise RSS captured by the AP in that room, and use it to effectively detect human movement in another room based solely on the antenna-wise RSS of whatever links are connected to the Wi-Fi AP in that room at that time. We train our method over AP-client links in one or more rooms. We test our approach in different rooms of varying size and with different furnishings, using different physical access points and clients with unseen links of varying length.

We show that our methods can detect human movement in larger rooms with over 73% and in smaller rooms with over 94% accuracy. This is without any calibration of models to the test rooms. The predictive performance achieved in the smaller rooms is comparable to what Lv, Man, Yang, Du, and Yu [Lv+18]—who use CSI data and models calibrated to the target environment—reported. We implement a state-of-the-art CSI-based method, R-TTWD [Zhu+17], which is designed to be robust to new environments, and apply it to our data. In a like-for-like comparison, our method performs comparably to R-TTWD when evaluated in the bigger rooms and clearly better when evaluated in the smaller rooms.

One of the main impediments to further development and adoption of device-free sensing capabilities is the paucity of publicly available data-sets for developing and comparing human sensing methods. We present the first publicly available CSI and (antenna-wise) RSS data-set that is annotated with ground truth information about human movement, presence, and absence in the area of interest [SCB20]. The data-set is the result of controlled experiments in four rooms, which we conducted over several weeks, and should be useful to anyone who is using Wi-Fi signals for device-free human presence detection or activity recognition.

The remainder of this chapter is organised as follows. Section 6.1 de-

scribes the four Wi-Fi testbeds, and sections 6.2 and 6.3 the physical and computational experiments. The results, which are discussed in section 6.4, show that our method is able to detect human movement in an unseen room from RSS data measured at each of the Wi-Fi access point's antennae with accuracy of 75.3% to 99.5% when using a single carefully selected link for training the model, and all the available links in the (held-out) test room to predict whether or not human movement occurred in the room.

6.1 Wi-Fi Testbeds

Figure 6.1 shows a schematic, drawn to scale, of the four testbed rooms. All four rooms are located in a university building which houses both teaching and research facilities. The two bigger rooms at the bottom of Figure 6.1, room G21 and G19, are computer science teaching labs on the ground floor of the building. They are located side by side, as shown in the figure. The wall at the bottom of the figure faces the building's atrium, and the rooms' opposing wall separates them from a larger teaching lab. To the left and right are two corridors. The bigger room (G21) seats 48 students and measures approximately 12 m × 7.6 m, while G19 only measures 8 m × 7.6 m and seats 23 students. In both rooms, projectors—two in G21 and one in G19—are mounted to the ceiling and aimed at the screens, which hang besides the door in the rooms' longest wall. In front of every chair there is a screen, keyboard, and mouse connected to a computer which is locked in a cage below the desktop.

The two smaller rooms, 128a and 260, are located on the first and second floor, respectively. The former, measuring approximately 4.2 m × 3.8 m, is flanked by the computer science departmental office on the left and a corridor on the right. On the other side of the wall opposite the door is the departmental tea kitchen. This room's main use is storage. There are documents—mostly bound theses—on the shelf opposite the door, a selection of historical computers and an overhead slide projector on the wall-mounted desk running along the left and bottom wall, and a ping-pong table is folded-up against the right wall. In the centre is a small table, surrounded by six

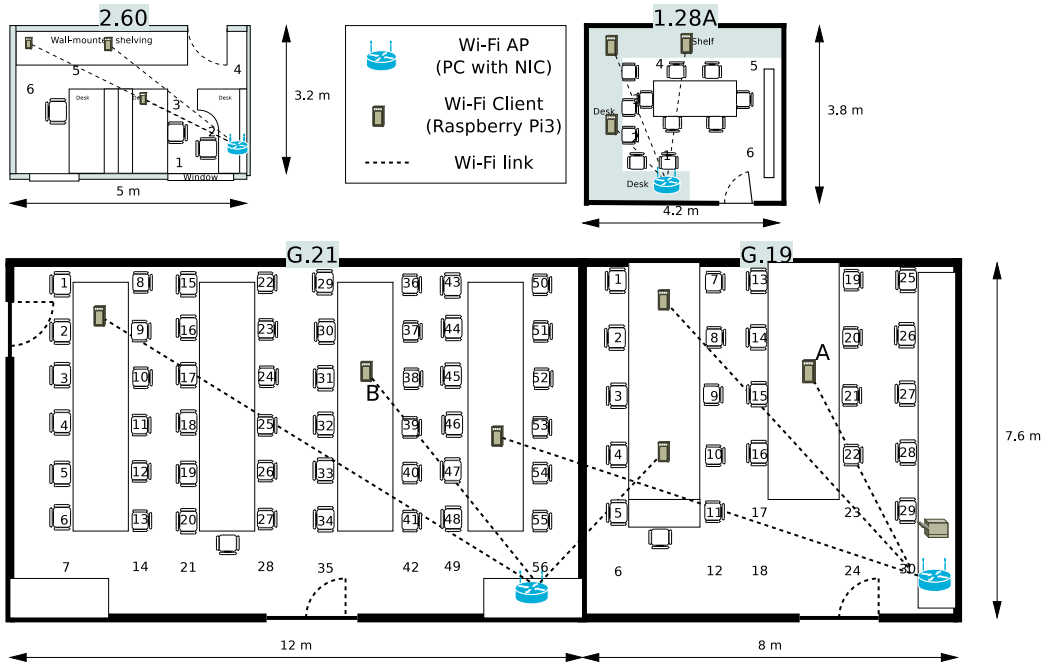


Figure 6.1: Schematic of the four rooms, drawn to scale, where we performed our experiments

chairs, which is sometimes used to hold meetings or interviews. The other room, number 260, measures approximately 5 m × 3.2 m, and is a three-desk office enclosed by a corridor on its left and top side, a smaller storage room on the right, and an open atrium at the bottom. Two of the desks stand back to back, their desktops separated by the shoulder-high metal-framed plywood boards that constitute their rear. While the desk in the middle is empty, apart from a few boxes, cables, paper, and other stationary utensils, the other desks each support a flat-screen monitor, keyboard, and mouse. Neither of the four rooms nor their furnishings have been modified or adapted for the sake of our experiments.

The Wi-Fi access points, illustrated by blue disks with two antennae in Figure 6.1, are regular PCs, each with an Intel 5300 network interface card (NIC) with three dipole antennae. The antennae are arranged in a plane that is (approximately) parallel to the desk supporting the PC, and orthogonal to its back, which runs parallel to the wall behind the desk and faces out into the room. Each PC, all of which run the Ubuntu Linux distribution, use the

`hostapd` program to instantiate a regular Wi-Fi AP. The APs only permit connections from wireless stations whose MAC address is listed in a whitelist, which contains the addresses of the clients depicted in Figure 6.1. The clients, which are depicted as grey rectangles in the figure, are Raspberry Pis. Specifically, we use the Raspberry Pi 3, model B+ in the bigger rooms, which comes with an integrated Cypress CYW43455 Wi-Fi chip that is connected to a PCB Proant Dual Band Niche antenna, while we use the Raspberry Pi 2 with an assortment of single-antenna Wi-Fi USB dongles in the smaller rooms. Most clients are attached with hook-and-loop fastening tape to a desktop, but the rightmost and leftmost client in room 128a and 260, respectively, are stuck onto the first shelf above the desktops—around 1.1 m above the floor—and, due to a lack of desk space, the client in the top left corner of 128a is taped to a cardboard filing box on the desk, rather than to the desk itself.

The dotted lines in Figure 6.1, then, represent Wi-Fi connections, meaning that all the clients connected via dotted lines to the same AP form a Wi-Fi network and can communicate with one another. To increase link diversity and the number of through-wall links, we modify the bigger rooms' Wi-Fi networks by connecting all clients to the other AP for half of the experiments. The physical separation between the ground, first, and second floor assures us that we can think of room 128a, room 260, and the two bigger rooms as three separate and independent testbeds. Conversely, while we cannot associate a client in room 128a with the AP in room 260 or vice versa, we can associate a client in G19 with the AP in G21 and vice versa. This allows us to include through-wall links in our experiments by connecting a client to an AP that is on the other side of a wall, something we could not do in the smaller rooms.

6.2 Acquiring Wi-Fi Signals for Evaluating Human Presence Detection Methods

To the best of our knowledge, no publicly available data-set exists that combines the antenna-wise RSS with human presence annotations. We

designed and conducted a set of experiments, in which Tim Creedon, our summer intern and myself acted as the occupants/subjects, to acquire such a data-set in the second half of 2018. The data-set is publicly available [SCB20] under a Creative Commons Attribution 4.0 International License[1].

Each experiment followed a basic protocol that stipulates three bouts of approaching and entering the room from a random direction, then moving to and remaining stationary at a random location for k minutes, before randomly moving about the room for k minutes. This terminates at another random location, where the occupant remains stationary for another k minutes, before moving to a door (randomly chosen, if there is more than one), exiting the room, and concluding the bout by departing in a random direction. The three bouts that constitute an experiment are preceded, separated, and followed by k-minute periods in which the room remains empty. The locations where the stationary and mobile presence examples take place are drawn from the locations indicated by the numbers in Figure 6.1. Ground truth—periods of emptiness, and stationary and mobile presence—is recorded by occupants by tapping the upcoming stage of the experiment in a smartphone app before initiating it, e.g., a tap on "Enter" before opening the door and entering the room, and a tap on "Door-Pos" after closing the door and prior to moving to the first position. The app, called "SensorLog" (now retired), records the tapped label (e.g., "Door-Pos") and time. Because AP and client clocks are synchronised to an NTP server over a wired network, we can use the recorded information to reconstruct the timeline of when a room was empty, and when it was occupied by a mobile or stationary occupant.

Note that although we did acquire examples of *stationary* presence, we shall not consider the problem of stationary presence detection in this chapter, for several reasons. The first is that if someone is present in a room at time t_i, then someone must have entered the room at some past time $t_j < t_i$—and entering a room implies some sort of *mobile* presence, which our method is designed to detect. This is analogous to how PIR-based motion detectors, which, as the name implies, do not detect human presence but motion, are most often used to automate lighting. Second, no living human can remain completely

[1] `https://creativecommons.org/licenses/by/4.0/`

motionless for extended periods of time without experiencing increasing discomfort, and eventual injury such as pressure ulcers (more commonly known as bedsores). Finally, we have conducted some pilot experiments with stationary and mobile presence, and found the former to be much more difficult than the second. Taken together, this means that having a method that reliably detects mobile presence is going to make detecting stationary presence easier.

We conducted six experiments in the smaller rooms, three experiments per room, and seven in the bigger rooms. For the experiments in the smaller rooms we set $k = 5$, and performed each example of stationary presence by sitting, then standing (or vice versa, determined randomly) for five minutes. The mobile presence examples were performed by walking at low, medium, or high speed, chosen at random for each bout. In the bigger rooms, we set $k = 3$, and modified the basic protocol by choosing at random which room the occupant was going to visit during a bout. We introduced changes in the physical environment for half of the experiments in the bigger rooms by putting a random selection of chairs on the desk prior to initiating the experiment. Furthermore, in addition to sitting and standing, some examples of stationary presence consisted of the occupant lying on the floor, and some mobile presence examples consisted of crawling. These additions are intended to simulate a sleeping or unconscious occupant, or a sneaky thief who is trying to evade detection.

If left to themselves the Wi-Fi networks do not generate much data, so during experiments we round-robin ping clients from their AP by sending one packet to the first client, waiting at most 0.1 s for a response, then moving on to the next client, and so on. The generated Wi-Fi packets are captured on the AP by a version of the `log_to_file` program that ships with the CSI tools [Hal+11], whose (C) source was modified to log each packet's source (the client) and destination (the AP) MAC address in addition to each antenna's RSS. The MAC addresses unambiguously identify which client transmitted captured packets.

This procedure resulted in an average of one hundred packets per second (100 Hz) per link in the smaller rooms, and 30 packets per second (30 Hz) per

link in the bigger rooms. The differences are mostly due to periods during which a link lost connectivity to the AP, which is more frequent in the bigger rooms than in the smaller rooms because the bigger rooms contain both through-wall and relatively long links, neither of which is the case in the smaller rooms.

6.3 Human Movement Detection and its Evaluation

Using data from pilot occupancy experiments in two smaller rooms, we performed various computational experiments and analyses to evaluate and select a small set of the most useful features, the size of the window along which they are extracted, and a machine learning (ML) algorithm which uses them to predict whether or not a human occupant is present in the room.

For each link, we separately extracted a set of time- and frequency-domain features along a 30 s sliding window with 50% (15 s) overlap from each link's antenna-wise RSS: eight univariate features—minimum, mean, maximum, standard deviation, skew, proportion of samples which exceed 90% of the maximum (PEAK), peak-power frequency (PPF), and spectral entropy—per antenna, the pairwise correlations between antennae, and two multivariate features—the eigenvalue of the first and second principal component (PCE1 and PCE2), and the largest and second-largest eigenvalue of the auto-correlation matrix (CORE1 and CORE2)—across all three antennae, for a total of 31 candidate input features.

From these features, we selected a small set that performs better than using all of them as follows. First, we applied correlation-based feature subset selection (CFS) [Hal98], using the WEKA [Hal+09, version 3.6.14] implementation, to identify subsets of features that correlate with the target but not each other. Because an algorithm's performance depends largely on its input features, we simultaneously assessed and evaluated various learning algorithms, namely logistic regression with L_1 (lasso) and L_2 (ridge) regularisation of varying strength, SVM with a radial basis function, LogitBoost, and

various multi-layer perceptron architectures, all of which were implemented in Python (version 3.7.3) using the sklearn library [Ped+11, version 0.20.2]. CFS most frequently selected PCE2, CORE2, the standard deviation from antenna A, the mean from antenna A and B, the PEAK from antenna B and C, and the PPF from all three antennae. Then, we analysed how sensitive the feature importance is when learning algorithms are trained with data from different days and links. We found that logistic regression with moderate ($C = 0.5$) L_1-regularisation performs comparably to, or even better than LogitBoost and Support Vector Machine (SVM), and that PCE2 conjoined with the three antennae's PPFs (PPF_A, PPF_B, and PPF_C) performs better than using all 29 features or relying on CFS alone to choose the best feature subset. Because regularisation is, as a rule, sensitive to features that are on different scales, we standardise each feature by subtracting its mean and dividing by its standard deviation, both of which are estimated from the training data.

Having found a promising combination of input features (PCE2, PPF_A, PPF_B, and PPF_C) and learning algorithm (L_1-regularised logistic regression), we assess how well it detects human movement in a given room (the test room) when using different approaches for selecting the training and test link(s). In particular, we assess the predictive performance when 1) trained with one link from one room and predicting from one or more (different) links in the test room (which may be the same room as the training link's room), 2) when trained with all the links in the training room and predicting with data from one or all links in the test room, 3) when trained on all the links that are not in the test room and predicting from one or all links in the test room, and 4) when trained on one link, selected among all the links that are not in the test room, and predicting from all the links in the test room. Each of these assessments corresponds to a different approach to train and deploy the system. In scenario 1) we estimate the accuracy when transferring models from one room with one link to a different set of links which may be in a different room. In scenario 2) we estimate accuracy when transferring from one room with multiple links to a different set of links in another room. In scenario 3) we estimate accuracy when transferring from multiple rooms, each with multiple links, to a different room. Finally, in scenario 4) we estimate

accuracy when selecting a single training link from multiple rooms, each with multiple links, and transfer that link's model to a different set of links in a different room.

Whenever we make predictions for more than one link we might (and usually do) end up with multiple predictions, one per link, all of which pertain to (approximately) the same (30 s) window. Clearly, presenting multiple distinct, possibly conflicting, predictions for the same period confers little informative value to users. To resolve this, we average (i.e., bag) the predicted probabilities (of human presence in the room) along a non-overlapping 30 s sliding window. This approach has the benefit of scaling well to any number of links, and not relying on information—such as its location or history—about the link. To balance the number of positive (someone is moving about the room) and negative (the room is empty) examples we down-sample each link's data, resulting in a data-set consisting of 2166 instances, of which 162 (7%) originated from room G21, 294 (14%) from room G19, 834 (39%) from room 128a, and 876 (40%) from room 260. The discrepancy between larger and smaller rooms is due to Wi-Fi connectivity issues, which naturally arise much more often with the longer and through-wall links that set the networks in the bigger rooms apart from those in the smaller rooms.

To compare our method with the state-of-the-art in CSI-based human presence detection, we implemented R-TTWD, a CSI-based method for human movement detection proposed by Zhu, Xiao, Sun, Wang, and Yang [Zhu+17] which we discussed in section 2.2. We apply R-TTWD to CSI data that correspond to the antenna-wise RSS data used by our method. The CSI data were acquired during the same experiments with the same Wi-Fi chipsets as the RSS data, such that each RSS sample corresponds to a CSI sample that was transmitted and received with the same Wi-Fi packet as the RSS sample.

6.4 Results and Discussion

Table 6.1 lists the median accuracy across single-link micro models, each of which is trained with data from a single link in the training room and whose predictions for each test link (all bar the training link) are used to calculate

Table 6.1: Median accuracy (%) when models trained on single links from the training room are tested on single links from the test room

	Test				
Train	128a	260	G19	G21	mean
128a	98.9	99.7	54.5	50.0	75.8
260	98.9	99.7	59.1	50.0	76.9
G19	97.8	100.0	79.5	62.5	85.0
G21	91.4	99.7	72.7	62.5	81.6
mean	96.8	99.8	66.5	56.2	79.8

the accuracy for each pair of training and test links. This is an indication of the accuracy that we should expect when using an arbitrary link for training and another for testing. Clearly, our method reliably discriminates between the presence of a mobile human and the empty room in smaller rooms even when trained with a through-wall link, i.e., one connecting the AP to a client in a different room. As a matter of fact, our method performs equally well in these rooms regardless of whether it was trained with data from another link in the same room, or with data from a link in the other small room. Even when trained with data from a link in a bigger room, our method detects human presence in a small room with over 91% accuracy. Although the worst train-/test-link pairs achieve a mere 35.7% and 39% accuracy in 128a and 260, respectively, the respective first quartiles are already at 68.3% and 77.3%. However, the results show that we cannot expect the same performance in a bigger room, particularly when the model is fitted to data from a room that is substantially smaller than the one it is deployed in. In this case, the predictive performance rarely and barely exceeds random guessing. Even when trained with data from another link in the same room, the accuracy of the median train-/test-link pair in G19 and G21 remains at 79.5% and 62.5%. Incidentally, there is no difference between the latter and the median accuracy achieved when predicting from a link in room G21 with a model trained in room G19, but there are 6.8 percentage points difference when the rooms' roles are reversed.

Table 6.2 lists the median accuracy across the single-link models from

Table 6.2: Median accuracy (%) across single-link models when averaging test-link predictions within each 30 s window

	Test				
Train	128a	260	G19	G21	mean
128a	98.6	100.0	78.5	72.0	87.3
260	98.7	100.0	77.9	70.7	86.8
G19	97.8	99.5	90.8	72.0	90.0
G21	92.5	99.5	87.2	69.0	87.0
mean	96.9	99.8	83.6	70.9	87.8

Table 6.1, but when each model's single-link predictions are averaged along a non-overlapping 30 s window across all the links in the test room. This is therefore an indication of the accuracy we can expect when using an arbitrary link for training a model, and all the links in the monitored room for testing. Again, we observe that our method excels in smaller rooms, but does not fare so well when deployed in bigger rooms. However, although the accuracy in the smaller rooms drops by a minuscule 0.2 to 0.5 percentage points (compared to Table 6.1) in some cases, there are clear increases in all other cases. On average, this approach improves the average single-link model's accuracy by 8 percentage points. The most important gains are those made in the bigger rooms. In room G19 and G21, median accuracy increases, on average by 17.1 and 14.7 percentage points, respectively. The biggest improvements emerge if a model is trained with a link from a small room and subsequently deployed in a bigger room, which is where the unaggregated single-link predictions from Table 6.1 perform the worst. This approach detects the presence of a mobile person in rooms that are different from the one used for training but of comparable or smaller dimensions with over 72% accuracy. Clearly, applying a trained model separately to whatever links are active on the system's network and aggregating those predictions over the desired time window is an effective way to improve the system's predictive performance, particularly when it is deployed in facilities that are bigger than the ones it was developed in.

An important consideration for human presence detection is the rate of false alarms or false positives—flagging an instance as human presence when

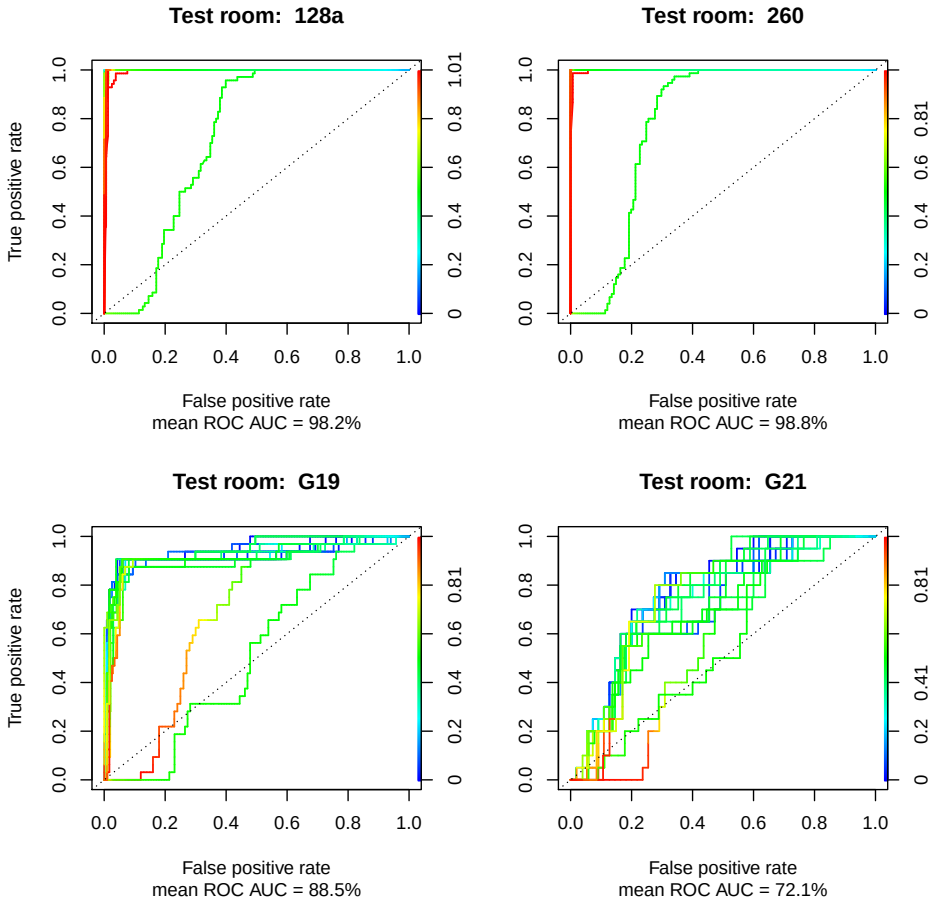


Figure 6.2: ROC curves of bagged single-link model predictions. Each curve corresponds to the bagged predictions across all of the test room's links, made by a model trained with a different training link.

Table 6.3: Median accuracy (%) when models trained on all links from the training room are applied to single test links

	Test				
Train	128a	260	G19	G21	mean
128a	-	99.7	63.0	50.0	70.9
260	98.9	-	63.6	50.0	70.8
G19	97.1	99.7	-	87.5	94.8
G21	82.7	90.8	66.7	-	80.1
mean	92.9	96.7	64.4	62.5	79.1

there is, in fact, no one present in the area of interest—and how they relate to the number of correctly detected presence events. Figure 6.2 illustrates this trade-off by way of receiver operator characteristic (ROC) curves. The figure shows one panel for each test room. Each curve corresponds to the bagged predictions from a model trained with data from a single link. These are the same predictions as are summarised in Table 6.2. The colour at a given point indicates the threshold that corresponds to the true and false positive rates at that point on the curve. In the smaller rooms, we observe that all but one training link offer excellent, and sometimes perfect, trade-offs between the true and false positive rate, which is reflected in the high mean ROC area under the curve (AUC) percentages. In the bigger rooms we see more diversity among training links. The top left corner of the ROC curves in G19 approximately corresponds to 90% true positives and fewer than 10% false alarms, but there are also two links that perform not much better than random. In G21 we see even more diversity among the links, and the curves are considerably closer to the diagonal. Even the better-performing links, which do considerably better than random guessing, only achieve approximately 80% true positives and 30% false alarms, with a mean ROC AUC of 72.1%. Fortunately, the majority of the curves in this room, too, stay well clear of the diagonal. This suggests that bad training links are in the minority. Which means that we might be able to combine multiple links into a single model to average out the bad, or use information from them to select only the good links.

Table 6.3 lists the median accuracy achieved by models trained with data

Table 6.4: Accuracy (%) when averaging the test-link predictions of models trained on all links from the training room

	Test				
Train	128a	260	G19	G21	mean
128a	-	100.0	78.5	69.3	82.6
260	98.7	-	78.5	72.0	83.1
G19	96.9	99.5	-	73.3	89.9
G21	78.9	93.5	56.4	-	76.3
mean	91.5	97.7	71.1	71.5	83.0

from all the links in the training room, whose predictions for each link in the test room are used to calculate the accuracy for that link. This, then, is an indication of the accuracy we should expect when using all available links in one room for training a model, and an arbitrary link in another room for testing it. On average, this approach performs slightly (0.7 points) worse than models trained with a single link on data from another single link. The worst results come from the model trained in G21. That model performs 6 percentage points worse than the corresponding single-link models from Table 6.1 in room G19, and nearly 9 points worse in 128a and 260. Still, there is also one prominent improvement. The model trained with the links from G19 achieves 87.5% median accuracy with a single link in room G21, 15.5 percentage points better than the average corresponding bagged single-link model from Table 6.2, and our best result for this room.

Table 6.4 lists the accuracy of models fitted to data from all the links in the training room whose predictions, separately made for each link in the test room, are averaged along a 30 s window, producing one prediction every 30 s. This gives us an idea of what to expect if we use the multi-link models whose performance is summarised in Table 6.3, but aggregate their predictions across all the links in the test room. This approach combines the multi-link models discussed in the previous paragraph with bagged multi-link predictions, which we found to be highly effective at improving performance, particularly in difficult rooms. It is therefore natural to expect an improvement on the results from Table 6.3 that is comparable to the one we achieved by bagging

the single-link model predictions. Yet this does not seem to be the case. We do see substantial improvements—comparable in magnitude to the ones achieved by bagging predictions for single-link models—when models trained in small rooms are evaluated in the big rooms. Still, the remaining results are comparable to, or much worse than the ones from Table 6.3, with one exception. The exception is the model trained in room G21, which achieves 90.8% accuracy when evaluated with the average link in room 260, but 93.5% when bagging its predictions for all the links in room 260. On average, we see an improvement of only 3.9 percentage points compared to Table 6.3. Perhaps the biggest disappointment is how the models trained in G19 and G21 perform when evaluated in the other big room. The model trained in G21 achieved 66.7% accuracy in room G19 with the average link, but when we aggregate the predictions from all six links the accuracy drops by 10.3 points to 56.4%. The model from room G19 looks even worse, falling from 87.5% accuracy in room G21, our best result in that room, to 73.3%. Despite the 14.2-point decrease this is our second-best result for room G21. There is however, as our next set of results shows, more than one way to achieve it.

Table 6.5 shows the performance when we consider all the links that are not in the test room as candidate training links, either indiscriminately using all of them for training the model, or selecting the link whose model performs best on the other candidate training links. This is an indication of how well different approaches to exploit all the available information from multiple training rooms perform when the learned model is transferred to a new room. Again, we achieve better performance in the smaller rooms, particularly when only relying on a single link in the test room. When it comes to multi-link deployments, we find that models fitted to all the links from the training rooms perform slightly (3.5 and 0.5 percentage points) better in smaller rooms than models only fitted to the best training link. Conversely, models fitted to the best training link outperform those fitted to all the training links by 7.3 and 4 percentage points in the big rooms, lifting accuracy from 81.9% and 69.3% to 89.2% and 73.3% in room G19 and G21, respectively.

The best link—the training link whose model outperforms the others on the other candidate training links—turns out to be the same link from room

Table 6.5: Accuracy (%) when models are trained with the best or all the links from the other rooms, and tested on the average (median) link or by averaging all test-link predictions

Links		Test room				
train	test	128a	260	G19	G21	mean
best	all	94.3	99.5	89.2	73.3	89.1
all	all	97.8	100.0	81.9	69.3	87.3
all	median	98.2	100.0	63.6	62.5	81.1

G19 in three out of the four test-room cases. The fourth case is when the test room, whose links are unavailable for training, is G19. Here, a link from G21 emerges as the most promising among all the candidate links. The selected training link from G19 connects that room's AP to the client marked by the letter "A" in Figure 6.1. It covers a distance of 5.1 m, making it the shortest link in the room. The selected link from room G21 connects the AP in that room to the client marked by the letter "B" in Figure 6.1. This link covers a distance of 5.8 m, only a little shorter than the median link (6.8 m) in G21.

How do these results compare with a state-of-the-art method for CSI-based human movement detection? Table 6.6 lists the results, corresponding to those given in Table 6.1 (i.e., the median accuracy), achieved by R-TTWD when it is trained with a single link in the training room and its predictions for each test link (all links bar the one used for training) are used to calculate the performance for each train- and test-link pair. While R-TTWD is better (by 0.7 to 17.9 percentage points) at detecting human movement in larger rooms when trained with data from smaller rooms, it performs comparably or worse in larger rooms when trained with data from another large room. It performs much worse (by 20 to over 30 percentage points) than our method in the two smaller rooms. On average, R-TTWD performs 18 percentage points worse with single training and test links than our approach.

Originally, R-TTWD was designed for a single training and test link, but we can easily adapt it to multiple test links by averaging the R-TTWD predictions across test links before combining, as described in the R-TTWD paper, the three CSI streams (one for each pair of transmitting and receiving

Table 6.6: Median accuracy (%) when R-TTWD is trained with single links from the training room and tested on single links from the test room

	Test				
Train	128a	260	G19	G21	mean
128a	63.2	60.0	59.8	67.9	62.7
260	63.2	60.0	59.8	67.9	62.7
G19	63.2	60.0	59.8	67.9	62.7
G21	63.2	60.0	58.7	53.8	58.9
mean	63.2	60.0	59.5	64.4	61.8

Table 6.7: Median accuracy (%) when averaging the single-link R-TTWD predictions prior to combining the three CSI streams via majority vote

	Test				
Train	128a	260	G19	G21	mean
128a	65.7	61.1	60.5	68.3	63.9
260	66.0	61.0	60.5	68.3	64.0
G19	66.0	61.1	60.5	68.3	64.0
G21	66.0	61.1	60.5	68.3	64.0
mean	65.9	61.1	60.5	68.3	64.0

antennae) via majority voting. The results when we adapt R-TTWD to multiple test links in this manner are shown in Table 6.7. These results show that averaging across test links does improve R-TTWD performance. Accuracy, relative to the unaggregated predictions, increases by 2.2 percentage points on average. We observe the biggest improvement when both training and testing in room G21, where median accuracy increases by 14.5 percentage points. Comparing this to the results in Table 6.2, which we achieved by aggregating our method's single-link predictions across test links in the same way, we find that our method clearly outperforms R-TTWD on all train-test room pairs. On average, the median bagged single-link models from Table 6.2 achieve 23.8, and the best training-link models from Table 6.5 25.1 percentage points higher accuracy than R-TTWD.

6.5 Conclusions

In this chapter, we presented our work on methods for detecting human movement from ambient Wi-Fi signal strength data. Unlike existing publications on device-free human sensing with Wi-Fi signals, which use the RSSI or CSI, our approach relies on the antenna-wise RSS measured by a regular Wi-Fi AP. We use just four features, namely the second principal component's eigenvalue and each antenna's peak-power frequency, which are extracted along a 30 s sliding window, as inputs to L_1-regularised logistic regression. We demonstrate the efficacy of our methods by cross-validating them in four different rooms of varying size and with different furnishings, using different physical Wi-Fi transceivers with unseen links of varying lengths.

In these tests, our method detects human movement in larger rooms (61 m^2 to 91 m^2) with accuracy above 75%, and in smaller rooms (area $\approx$ 16 m^2) with accuracy of 96% or above. We achieve these results by training a model to detect human movement in one room with RSS data from a single, carefully selected, link, and then use it, without any calibration, to detect human movement in another room based on the RSS from whatever links are available in that room. The results in the smaller rooms are comparable to those reported for state-of-the-art CSI-based methods, but unlike these

methods, our method does not need calibration to the target room nor any knowledge about the links whose Wi-Fi signals are the basis for detecting human movement. Our approach performs comparably or better on our dataset than R-TTWD, a state-of-the-art method for human movement detection based on CSI specifically designed to be robust to environmental changes. To foster further research and progress in the field we make our data, consisting of annotated (antenna-wise) RSS and CSI signals, publicly available [SCB20], in the hope that it will be a useful resource for those readers who are working on device-free human sensing with Wi-Fi signals.

Chapter 7

Conclusions & Future Work

The evidence presented in this thesis and the conclusions emanating from them, which are summarised in this chapter, support the thesis statement:

> Micro models and domain knowledge can improve human sensing methods. In particular:
>
> - person-specific micro models can improve the subject-dependent performance of methods for human activity recognition from wearable sensor data;
> - link-specific micro models, either individually or in ensemble, can improve the predictive performance of device-free wireless human presence detection methods;
> - domain knowledge encoded in expert hierarchies simplifies multi-class human activity recognition models while maintaining predictive performance.

We looked at micro (and macro) models and the use of domain knowledge in the form of activity hierarchies for human activity recognition (HAR) from wearable inertial sensor data. We further showed how to exploit micro models for device-free human movement detection from ambient Wi-Fi signal data. Our results show that in these settings micro models and domain knowledge in the form of expert hierarchies can indeed improve human sensing methods. In this chapter, we summarise the main conclusions (in section 7.1) before

ending the thesis (in section 7.2) with suggestions for directions of future work in these areas.

7.1 Conclusions

7.1.1 Human Activity Recognition for Emergency First Responders

In chapter 3 we present a method for HAR that is designed for the dynamic activities and environments in which emergency first responders operate. The proposed method extracts a set of handpicked time- and frequency-domain features from inertial measurement unit (IMU) signals. The features are fed to a gradient boosted ensemble of decision trees (GBT) which we specifically design and tune for this purpose. We address the uncertainty about what the appropriate activities of interest are by considering as many fine-grained activities as practically feasible, and then use domain knowledge to combine them into groups of coarser activities. We use the resulting four HAR problems to tune four machine learning algorithms—among them our GBT–for their expected subject-independent performance. We compare the algorithms' subject-dependent and -independent performance, which is estimated via eleven-fold cross-validation (CV) and leave-one-subject-out cross-validation (LOSO CV) (both across all users), respectively. Our results show that our GBT clearly outperforms the other algorithms on both subject-dependent and -independent performance for all but one of the four problems, viz. fall detection.

7.1.2 Subject-Dependent and -Independent Human Activity Recognition with Micro and Macro Models

In chapter 4 we investigate person-specific models (PSMs) as a way to boost predictive HAR performance for known users. PSMs are user-specific micro models that are applied to all future instances from their user. The downside of PSMs is that it is not obvious how they can be used to make predictions

for unknown users. We show that the issue can be circumvented, and the PSMs used to obtain predictions for users not represented in the training data, by combining them into an ensemble of person-specific models (EPSM). One of the main advantages of EPSMs are that they can incorporate data from new users without accessing data from other users. This liberates human sensing system operators from the technical, administrative, and legal burden of storing—and thus controlling—an ever-growing stash of potentially intimate data about all its past and present users. We empirically evaluate these methods, including three different flavours of EPSMs, against the (macro) person-independent model (PIM) that dominates the literature. These computational experiments include seven benchmark data-sets and four HAR inference algorithms, including the GBT from chapter 3. To account for correlations within data-sets, we subject the results to a sophisticated statistical analysis with generalised linear mixed-effects models (GLMMs).

The analysis shows that when tested on known users, the odds of a correct classification are 44% better with PSMs than with the corresponding PIM. Here too, the GBT we presented in chapter 3 clearly outperforms the other algorithms we compare it against both on subject-dependent and on subject-independent performance, confirming the soundness of the process that led to it. What is more, when we compare our results for the six publicly available data-sets we used to the state-of-the-art results reported in the literature, we find that our GBT achieves subject-independent performance comparable to the state-of-the-art for five data-sets, and comparable and better subject-dependent performance than the state-of-the-art on one and four data-sets, respectively. Our results also show that combining PSMs into an EPSM dramatically improves the otherwise abysmal subject-independent performance of PSMs, raising it to within a few percentage points of the accuracy achieved by the corresponding PIM. These results are evidence that micro models indeed can improve the predictive performance of human sensing methods, in this case the subject-dependent performance of HAR methods with wearable inertial sensor data.

7.1.3 Human Activity Recognition with Expert Hierarchies

In chapter 5 we return to domain knowledge as a source of information for the efficient development of effective HAR capabilities. We propose to leverage domain knowledge encoded in the intuitive form of activity hierarchies to build expert hierarchies. Expert hierarchies are one way to make a multi-class classification problem amenable to binary classification algorithms by decomposing it into a set of binary classification problems. What sets them apart from other multi-class decomposition methods is not only lower computational complexity at training and prediction time but also, and perhaps more importantly, the independence of their constituent binary classifiers. This independence means that each of the binary classifiers constituting an expert hierarchy can be developed and tuned in isolation from the rest of the system, which holds potential for streamlining many parts of the HAR development process, most importantly the particularly time-consuming process of data annotation. It may also hold promise for combining data from disparate HAR data-sets whose activities of interest overlap only partially, an idea we shall discuss in more detail in subsection 7.2.2.

We compare five expert hierarchies, each inspired by either an end-user or engineering perspective of the original 17-class problem from chapter 3, against other approaches to multi-class classification. Our comparison includes not only four other multi-class decompositions, namely one-versus-all (OVA), one-versus-one (OVO), error-correcting output codes (ECOCs), and ensembles of nested dichotomies (ENDs), but also the direct multi-class formulation of the problem. By applying each of these methods not only to the original 17-class HAR problem, but also to the binary problems induced by the topmost dichotomy of two of the five expert hierarchies, we demonstrate that favourable performance on the multi-class problem does not imply good performance on any or all of the binary problems.

In particular, we find that OVO, which tends to compare favourably against other multi-class decomposition methods in the literature and does quite well on our multi-class problem too, does not perform particularly

well (or badly) on our two binary problems. The same holds for our expert hierarchies, which perform comparably to the standard OVA approach to multi-class classification. This shows that expert hierarchies, which encode domain knowledge about human activities, are indeed a viable alternative to other multi-class HAR approaches. We further find that there is clearly less variance between expert hierarchies themselves than between the other methods. Which suggests that a quest to find the best expert hierarchy is not the most cost-efficient way to improve predictive HAR performance. Expert hierarchies thus offer a simple and intuitive way to leverage domain knowledge for HAR systems that incurs no penalty on predictive performance, but instead imparts other benefits such as modular design and development. These results are evidence that domain knowledge encoded in expert hierarchies simplify multi-class classification models, while maintaining predictive performance.

7.1.4 Detecting Human Movement from Ambient Wi-Fi Signal Strength with Micro Models

In chapter 6 we present the first method for human movement detection from antenna-wise Wi-Fi received signal strength (RSS) data. We exploit micro models, each of which corresponds to a particular Wi-Fi link in this application, to build link-specific models. These single-link models are analogous to the PSMs from chapter 4 insofar as each model is trained with data from a specific data source, but differ in that we exploit the diversity between links to find a link-specific micro model that performs well on data from other links. We further consider two types of multi-link macro models—one trained with data from a single (training) room, the other with data from all but one (the test) room—as well as R-TTWD [Zhu+17], a state-of-the-art method based on channel state information (CSI). What primarily sets our method for human movement detection data apart from others is that we explicitly target ambient Wi-Fi signals. The word "ambient" here is meant to imply that we aim to exploit existing Wi-Fi infrastructure, without assuming any control over, or knowledge about, the number or location of the Wi-Fi clients whose transmissions we exploit.

Unfortunately, there are no publicly available data-sets we could use to benchmark our methods. To evaluate our approach, we created a data-set of RSS and CSI signals which we acquired from four testbeds, with our summer intern and myself acting as the occupants in the experiments. Each testbed was set up in a different room, and consisted of a regular Wi-Fi access point (AP) and three clients. The two smaller rooms—a two-person research office and a small storage/meeting room—are located on different floors of the building, while the larger rooms are two teaching labs on the ground floor, separated only by a thin wall. We put the wall to use by considering network configurations with one or two through-wall links to a client on the other side. We do our bit to remedy the lack of publicly available data-sets with Wi-Fi signals for human presence detection by publishing ours in an open online repository [SCB20].

We evaluate our single- and multi-link models, as well as R-TTWD on this data-set with CV procedures that split data into train and test sets according to the room or link. Our results show that the best approach is to fit link-specific micro models to the links from multiple (training) rooms, select the one that performs best on the other links in those same training rooms, use it to make predictions from whatever links are transmitting in the test room, and average those predictions along a sliding window. This method detects human movement with 73.3% and 89.2% accuracy in the two larger rooms, and 94.3% and 99.5% in the smaller rooms. For comparison, our implementation of R-TTWD achieves at best 66% accuracy in the smaller and 68% in the larger rooms. These results show that we can automatically detect human movement from ambient Wi-Fi received signal strength in entirely new rooms without making assumptions about the room's layout or Wi-Fi network. They also are further evidence that micro models indeed can improve predictive performance of human sensing methods, in this case the cross-facility performance of a method for human presence detection from ambient Wi-Fi signals.

7.2 Future Work

In this thesis, we have focused on two human sensing applications, namely wearable HAR and device-free presence detection. However, both micro models and expert hierarchies are general techniques that can be useful not only for a wide variety of human sensing problems but also in a wide range of other domains. Nevertheless, exploring these techniques in other applications and domains is only one avenue of future work. In this section, we outline ideas of how micro models and ensembles of micro models could further be improved, discuss how expert hierarchies could be extended and exploited, and list ideas that could further improve device-free human presence detection, particularly for unseen facilities and links.

7.2.1 Micro and Macro Models

To apply micro models to data from unseen entities, such as a new users or wireless links, we have considered either a single carefully selected (link-specific) micro model or ensembles that combine all the (person-specific) micro models. Our results show that weighted EPSMs significantly outperform their unweighted counterparts, and that some link-specific models perform exceptionally bad on new links while others perform very well. This suggests that the difference among micro models' performances contains information that could be exploited to improve ensembles' ability to generalise to unseen entities. We could, for example, exclude the worst micro models—e.g., those with a serious risk of performing no better than random guessing—from ensembles of micro models such as EPSMs. Other machine learning techniques for improving ensembles (such as stacking and minimising the correlation among models) could also be considered. Or we could build multiple ensembles of micro models at intermediate (meso) levels—e.g., according to first responders' level of experience, or according to link length or location. These meso-level ensembles could then be aggregated in macro-level ensembles. Different weighting schemes such as the performance on other (meso) ensembles' training data—could be considered.

Another way to improve ensembles of micro models could be to separately

optimise individual micro models, which of course ought to also improve performance for the model's corresponding data-generating entity (e.g., user or link). The optimisation could consist of model selection, hyper-parameter tuning, or both. In theory, ensembles perform better if their constituent models are more diverse—i.e., issue predicitons that are less correlated. We can expect that increasing model flexibility also increases their variance, which ought to be a good thing. However, this only applies if models on average perform better than random. It therefore might be beneficial, or even necessary, to include only those micro models in an ensemble that perform significantly better than random.

We have shown that person-specific micro models (PSMs) tend to achieve better subject-dependent performance than the corresponding person-independent macro model (PIM), and that a PIM outperforms the corresponding EPSM. However, this does not preclude that, given a sufficiently large number of users, there may be a point at which a PIM achieves better subject-dependent performance than PSMs, or, alternatively, an EPSM better subject-independent performance than the corresponding PIM. To properly test these hypotheses, we would need several data-sets, each with a large number (ideally, hundreds or even thousands) of users. Unfortunately, few HAR data-sets cover more than 20 users, and we are not aware of any that cover anywhere near the required number. Fortunately, expert hierarchies might offer a way to circumvent this issue.

7.2.2 Expert Hierarchies

Expert hierarchies might offer a path to combine HAR data from multiple data-sets, whose activities may overlap only partially, into a single HAR data-set that is not only much bigger but also more diverse. We could design an activity hierarchy that covers all the activities across all data-sets. The hierarchy's topmost dichotomy (e.g., "stationary" versus "mobile" acivities) effectively constitutes a single data-set that is not only much bigger, but also more diverse in terms of users and IMUs. The combined data-set could then be used not only to investigate the hypotheses from the previous paragraph,

but also to develop and evaluate methods for transferring HAR models from the activities, users, and IMUs in one subset of the original data-sets to those in another. There are, however, several open issues. For instance, we would have to find a stopping criterion to avoid descending further down the hierarchy when the deepest level has been reached that matches both the model's and the target data-set's activities of interest. While the threshold we formulated in chapter 5 can be used for this, there is no reason to believe that it will achieve the intended result, and other approaches are possible and just as plausible.

As we have pointed out, one of the main advantages of expert hierarchies over OVA—and other non-hierarchical multi-class decomposition methods—is that the independence of their constituent binary classifiers permits tuning each classifier independently. In chapter 5, we have seen expert hierarchies perform comparably to OVA without further tuning. We likely could improve the performance of expert hierarchies further by separately performing model selection and hyper-parameter tuning for each of its constituent classifiers.

Expert hierarchies do not have to be confined to classification, but could be extended to regression (predicting a number rather than a category). We could do this, for example, with a top-level binary classifier that labels observations as either having "low" or "high" values of the target variable, then use a separate regression model for each of the two branches that predicts the precise quantity. Conceptually, this is similar to Gaussian Mixture Models, but with sub-populations constrained or determined by expert knowledge, rather than learned from the data. With continuous expert hierarchies, as we might call them, the structure of the hierarchy—e.g., whether we have only "high" and "low," or "low," "medium," and "high" categories—is not the only thing that can be determined by domain knowledge. There is also the question of where, precisely, the threshold lies that separates "low" and "high." This threshold could be determined based either on domain knowledge or on data, and it is not obvious whether one approach is preferable over the other.

7.2.3 Device-free Human Sensing with Wi-Fi Signals

We could use a support set of empty-room data to scale the data from each wireless link relative to its empty-room state prior to applying the method presented in chapter 6. In subsection 2.2, we discussed RASID [KSY12], a RSS-based device-free presence detection which uses micro models that are calibrated with 2 min of data acquired from the target link while the area of interest is vacant. Combining this idea with our approach could go a long way towards improving presence detection.

Our presence detection method could be evaluated with multiple occupants and successively extended to people counting—i.e., estimating how many occupants are present in the area of interest. Expert hierarchies might be useful here, too. For example, we could use a simple hierarchy that consists of a model that discriminates between human absence and presence at the top, and one that estimates the number of occupants which descends from the "human presence" branch.

We also could evaluate our methods in a wider variety of even more realistic Wi-Fi constellations and traffic profiles and over longer periods of time. These would extend beyond three simultaneous links and include generating traffic that is more in line with the more stochastic behaviour associated with everyday tasks such as video streaming, web browsing, or machine-to-machine communications. In addition to further testing our approach, this would also offer an interesting opportunity to exploit Wi-Fi network traffic patterns and combine them with signal-level data.

Our approach to device-free human presence detection has been designed to run on Wi-Fi APs, as opposed to Wi-Fi clients, because this only requires access to the APs and makes no assumptions about the number and location of connected clients. It would be interesting to see whether our approach works with data captured by a Wi-Fi node in "monitoring" mode (also called a Wi-Fi "sniffer") rather than an AP. Deploying presence detection on Wi-Fi sniffers has the advantage that a sniffer receives *all* Wi-Fi signals that are transmitted on the monitored Wi-Fi channel, whereas an AP only receives signals from its associated clients. By rapidly hopping between channels,

sniffers can further extend the number of monitored channels and thus the number of wireless links whose signals are available for human sensing.

Similarly, it would be interesting to see whether or not we can detect human movement (or presence) in the room in which the Wi-Fi client, rather than the AP, is located. Deploying human presence detection on Wi-Fi clients, rather than APs, might have advantages, too. Users could deploy small Wi-Fi clients in the areas they want monitored. Other work has shown that the effects of human presence on Wi-Fi signals rapidly decrease with increasing distance between person and link. This indicates that integrating the outputs of client-centred detection algorithms with their client locations could be an effective way to achieve presence detection whose area of interest, spatial resolution, and accuracy can be configured or tuned through the placement and density of client nodes.

Acronyms

ADL activity of daily living

AP access point

CFS correlation-based feature subset selection

C.I. confidence interval

CNN convolutional neural network

CSI channel state information

CV cross-validation

ECOC error-correcting output code

EH expert hierarchy

END ensemble of nested dichotomies

EPSM ensemble of person-specific models

DCNN deep convolutional neural network

DNN deep neural network

DT decision tree

GBT gradient boosted ensemble of decision trees

GLMM generalised linear mixed-effects model

HAR human activity recognition

IMU inertial measurement unit

kNN k-Nearest Neighbours

LOSO CV leave-one-subject-out cross-validation

LSTM long short-term memory

MAE mean absolute error

MDM multi-class decomposition method

NIC network interface card

OVA one-versus-all

OVO one-versus-one

PC principal component

PCA principal component analysis

PCTL percentile

PGA personalisation-generalisation approach

PIM person-independent model

PPF peak-power frequency

PSM person-specific model

RF radio frequency

ROC receiver operator characteristic

RSS received signal strength

RSSI received signal strength indicator

SD standard deviation

SE standard error

SVM Support Vector Machine

WEPSM$_{\mathbf{bf}}$ baseline-feature-weighted ensemble of person-specific models

WEPSM$_{\kappa}$ κ-weighted ensemble of person-specific models

References

[AF18] Fayez Alharbi and Katayoun Farrahi. "A Convolutional Neural Network for Smoking Activity Recognition". In: *International Conference on e-Health Networking, Applications and Services* (Ostrava, CZE). Healthcom. IEEE, Sept. 2018, pp. 1–6. DOI: 10.1109/HealthCom.2018.8531148.

[AFH15] Dina Bousdar Ahmed, Korbinian Frank, and Oliver Heirich. "Recognition of Professional Activities with Displaceable Sensors". In: *Vehicular Technology Conference*. VTC. IEEE, Sept. 2015, pp. 1–5. DOI: 10.1109/VTCFall.2015.7391112.

[AG19] Mubarak G. Abdu-Aguye and Walid Gomaa. "Competitive Feature Extraction for Activity Recognition based on Wavelet Transforms and Adaptive Pooling". In: *International Joint Conference on Neural Networks* (Budapest, HUN, July 14–19, 2019). IJCNN. IEEE, July 2019, pp. 1–8. DOI: 10.1109/IJCNN.2019.8852299.

[AK13] Fadel Adib and Dina Katabi. "See through Walls with WiFi!" In: *SIGCOMM Computer Communication Review* 43.4 (Aug. 2013), pp. 75–86. ISSN: 0146-4833. DOI: 10.1145/2534169.2486039.

[ASS00] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers". In: *Journal of Machine Learning Research* 1.2 (Dec. 2000).

[Bañ+14] Oresti Baños, Rafael Garcia, Juan A. Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. "mHealthDroid: A Novel Framework for Agile

Development of Mobile Health Applications". In: *International Workshop on Ambient Assisted Living* (Belfast, GBR, Dec. 2–5, 2014). IWAAL. Springer, 2014. DOI: 10.1007/978-3-319-13105-4_14.

[Bat+15] Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. "Fitting Linear Mixed-Effects Models Using lme4". In: *Journal of Statistical Software* 67.1 (2015). DOI: 10.18637/jss.v067.i01. URL: https://cran.r-project.org/web/packages/lme4.

[Bau+12] Miguel Ángel Bautista, Sergio Escalera, Xavier Baró, Petia Radeva, Jordi Vitrià, and Oriol Pujol. "Minimal design of error-correcting output codes". In: *Pattern Recognition Letters* 33.6 (2012). ISSN: 0167-8655. DOI: 10.1016/j.patrec.2011.09.023.

[BBS14] Andreas Bulling, Ulf Blanke, and Bernt Schiele. "A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors". In: *Computing Surveys* 46.3 (Jan. 2014). DOI: 10.1145/2499621.

[BG17] Ramon F. Brena and Enrique Garcia-Ceja. "A crowdsourcing approach for personalization in human activities recognition". In: *Intelligent Data Analysis* 21.3 (2017), pp. 721–738. ISSN: 1088467X. DOI: 10.3233/IDA-170884.

[BI04] Ling Bao and Stephen S. Intille. "Activity Recognition from User-Annotated Acceleration Data". In: *International Conference on Pervasive Computing* (Linz/Vienna, AUT, Apr. 21–23, 2004). PERVASIVE. 2004. DOI: 10.1007/978-3-540-24646-6_1.

[BLL98] François Bergeron, Gilbert Labelle, and Pierre Leroux. *Combinatorial species and tree-like structures*. Cambridge University Press, 1998.

[Bou+16] Remco R. Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, and David Scuse. *WEKA Manual*. Aug. 2016.

[BS09] Ulf Blanke and Bernt Schiele. "Daily Routine Recognition through Activity Spotting". In: *International Symposium on Location- and Context-Awareness* (Tokyo, JPN, May 7–9, 2009). LoCA. Springer, 2009. DOI: 10.1007/978-3-642-01721-6_12.

[Bui+13] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. "API design for machine learning software. Experiences from the scikit-learn project". In: *European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases*. Workshop on Languages for Data Mining and Machine Learning (Prague, CZE, Sept. 23–27, 2013). ECML/PKDD. 2013.

[Cat+15] Cagatay Catal, Selin Tufekci, Elif Pirmit, and Guner Kocabag. "On the Use of Ensemble of Classifiers for Accelerometer-Based Activity Recognition". In: *Applied Soft Computing* 37.C (Dec. 2015), pp. 1018–1022. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2015.01.025.

[CH92] Gregory F. Cooper and Edward Herskovits. "A Bayesian method for the induction of probabilistic networks from data". In: *Machine Learning* 9.4 (Oct. 1992), pp. 309–347. DOI: 10.1007/bf00994110.

[Cha+13] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R. Millán, and Daniel Roggen. "The Opportunity challenge. A benchmark database for on-body sensor-based activity recognition". In: *Pattern Recognition Letters* 34.15 (2013): *Smart Approaches for Human Action Recognition*. DOI: 10.1016/j.patrec.2012.12.014.

[Coh95] William W. Cohen. "Fast Effective Rule Induction". In: *International Conference on Machine Learning* (Tahoe City, USA, July 9–12, 1995). Ed. by Armand Prieditis and Stuart Russell.

ICML. Morgan Kaufmann, 1995, pp. 115–123. ISBN: 1-55860-377-8. DOI: 10.1016/B978-1-55860-377-6.50023-2.

[CS02] Koby Crammer and Yoram Singer. "On the Algorithmic Implementation of Multiclass Kernel-Based Vector Machines". In: *Journal of Machine Learning Research* 2 (Mar. 2002). ISSN: 1532-4435.

[CW03] Yixin Chen and James Z. Wang. "Support vector learning for fuzzy rule-based classification systems". In: *Transactions on Fuzzy Systems* 11.6 (Dec. 2003). ISSN: 1063-6706. DOI: 10.1109/TFUZZ.2003.819843.

[CX15] Yuqing Chen and Yang Xue. "A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer". In: *International Conference on Systems, Man, and Cybernetics* (Kowloon, CHN, Oct. 9–12, 2015). IEEE. Oct. 2015, pp. 1488–1492. DOI: 10.1109/SMC.2015.263.

[CY18] Heeryon Cho and Sang Min Yoon. "Divide and Conquer-Based 1D CNN Human Activity Recognition Using Test Data Sharpening". In: *Sensors* 18.4 (2018). ISSN: 1424-8220. DOI: 10.3390/s18041055.

[DB95] Thomas G. Dietterich and Ghulum Bakiri. "Solving multiclass learning problems via error-correcting output codes". In: *Journal of artificial intelligence research* 2 (1995). DOI: 10.1613/jair.105.

[DFK05] Lin Dong, Eibe Frank, and Stefan Kramer. "Ensembles of Balanced Nested Dichotomies for Multi-class Problems". In: *European Conference on Principles and Practice of Knowledge Discovery in Databases*. ECML/PKDD. Berlin, Heidelberg: Springer, 2005. ISBN: 3-540-31665-5. DOI: 10.1007/11564126_13.

[DG17] Dheeruo Dua and Casey Graff. *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, 2017. URL: http://archive.ics.uci.edu/ml.

[Erm+08] Miikka Ermes, Juha Pärkkä, Jani Mäntyjärvi, and Ilkka Korhonen. "Detection of Daily Activities and Sports with Wearable Sensors in Controlled and Uncontrolled Conditions". In: *Transactions on Information Technology in Biomedicine* 12.1 (Jan. 2008). ISSN: 1089-7771. DOI: 10.1109/TITB.2007.899496.

[Est+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *International Conference on Knowledge Discovery and Data Mining*. KDD. AAAI Press, 1996, pp. 226–231.

[Fer+20] A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano. "On the Personalization of Classification Models for Human Activity Recognition". In: *IEEE Access* 8 (2020), pp. 32066–32079. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2973425.

[FG15] Giancarlo Fortino and Raffaele Gravina. "Fall-MobileGuard: a Smart Real-Time Fall Detection System". In: *International Conference on Body Area Networks* (Sydney, AUS, Sept. 28–30, 2015). BODYNETS. EAI and CREATE-NET. ACM, Dec. 14, 2015. DOI: 10.4108/eai.28-9-2015.2261462.

[FK04] Eibe Frank and Stefan Kramer. "Ensembles of Nested Dichotomies for Multi-class Problems". In: *International Conference on Machine Learning*. ICML. New York, NY, USA: ACM, 2004. ISBN: 1-58113-838-5. DOI: 10.1145/1015330.1015363.

[Fox97] John Fox. *Applied regression analysis, linear models, and related methods*. Sage, 1997. ISBN: 0-803-94540-X.

[Fra+14] Korbinian Frank, Estefania Munoz Diaz, Patrick Robertson, and Francisco Javier Fuentes Sánchez. "Bayesian recognition of safety relevant motion activities with inertial sensors and barometer". In: *IEEE/ION Position, Location and Navigation Symposium*. PLANS. May 2014, pp. 174–184. DOI: 10.1109/PLANS.2014.6851373.

[Fri96] Jerome H. Friedman. *Another Approach to Polychotomous Classification*. Tech. rep. Stanford University, Oct. 1996. URL: http://statweb.stanford.edu/~jhf/ftp/poly.pdf.

[Gal+11] Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. "An overview of ensemble methods for binary classifiers in multi-class problems. Experimental study on one-vs-one and one-vs-all schemes". In: *Pattern Recognition* 44.8 (2011). ISSN: 0031-3203. DOI: 10.1016/j.patcog.2011.01.017.

[GH06] Andrew Gelman and Jennifer Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2006. ISBN: 0-521-68689-X.

[Guo+17] Linlin Guo, Lei Wang, Jialin Liu, Wei Zhou, Bingxian Lu Tao Liu, Guangxu Li, and Chen Li. "A novel benchmark on human activity recognition using WiFi signals". In: *International Conference on e-Health Networking, Applications and Services*. Healthcom. Oct. 2017, pp. 1–6. DOI: 10.1109/HealthCom.2017.8210783.

[Hal+09] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA Data Mining Software: An Update". In: *SIGKDD Explorations Newsletter* 11.1 (Nov. 2009), pp. 10–18. ISSN: 1931-0145. DOI: 10.1145/1656274.1656278.

[Hal+11] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. "Tool Release: Gathering 802.11n Traces with Channel State Information". In: *Computer Communication Review* 41.1 (Jan. 2011). ISSN: 0146-4833. DOI: 10.1145/1925861.1925870.

[Hal98] Mark A. Hall. "Correlation-based Feature Subset Selection for Machine Learning". PhD thesis. University of Waikato, 1998.

[He+15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: *Transactions on Pattern Analysis and Machine*

Intelligence 37.9 (Sept. 2015), pp. 1904–1916. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2015.2389824.

[HFF12] José Hernández-Orallo, Peter Flach, and Cèsar Ferri. "A Unified View of Performance Metrics. Translating Threshold Choice into Expected Classification Loss". In: *Journal of Machine Learning Research* 13 (Oct. 2012).

[HHP16] Nils Y. Hammerla, Shane Halloran, and Thomas Plötz. "Deep, Convolutional, and Recurrent Models for Human Activity Recognition Using Wearables". In: *International Joint Conference on Artificial Intelligence*. IJCAI. Apr. 2016.

[HL02] Chih-Wei Hsu and Chih-Jen Lin. "A comparison of methods for multiclass support vector machines". In: *Transactions on Neural Networks* 13.2 (Mar. 2002), pp. 415–425. ISSN: 1941-0093. DOI: 10.1109/72.991427.

[HT98] Trevor Hastie and Robert Tibshirani. "Classification by pairwise coupling". In: *The Annals of Statistics* 26.2 (Apr. 1998). DOI: 10.1214/aos/1028144844.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. 2nd ed. Springer, 2009. ISBN: 0-387-84857-6. DOI: 10.1007/978-0-387-84858-7.

[JFY09] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. "Cutting-Plane Training of Structural SVMs". In: *Machine Learning* 77.1 (Oct. 2009), pp. 27–59. ISSN: 0885-6125. DOI: 10.1007/s10994-009-5108-8.

[JOP+01] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–. URL: https://www.scipy.org.

[Jor+19] Artur Jordao, Antonio Carlos Nazare Jr., Jessica Sena, and William Robson Schwartz. "Human Activity Recognition Based on Wearable Sensor Data. A Standardization of the State-of-the-Art". In: *CoRR* (Feb. 2019). arXiv: 1806.05226v3 [cs.CV].

[Jos+10] Sandeep J. Joseph, Kelly R. Robbins, Wensheng Zhang, and Romdhane Rekaya. "Comparison of Two Output-Coding Strategies for Multi-Class Tumor Classification Using Gene Expression Data and Latent Variable Model as Binary Classifier". In: *Cancer Informatics* 9 (Jan. 1, 2010). DOI: 10.4137/CIN.S3827.

[JY15] Wenchao Jiang and Zhaozheng Yin. "Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks". In: *International Conference on Multimedia* (Brisbane, AUS). MM. New York, NY, USA: ACM, 2015, pp. 1307–1310. ISBN: 1-4503-3459-8. DOI: 10.1145/2733373.2806333.

[Kar+06] Dean M. Karantonis, Michael R. Narayanan, Merryn J. Mathie, Nigel H. Lovell, and Branko G. Celler. "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring". In: *Transactions on Information Technology in Biomedicine* 10.1 (Jan. 2006). DOI: 10.1109/TITB.2005.856864.

[Kia+17] Sanaz Kianoush, Stefano Savazzi, Federico Vicentini, Vittorio Rampa, and Matteo Giussani. "Device-Free RF Human Body Fall Detection and Localization in Industrial Workplaces". In: *Internet of Things Journal* 4.2 (Apr. 2017). ISSN: 2327-4662. DOI: 10.1109/JIOT.2016.2624800.

[KSY12] Ahmed E. Kosba, Ahmed Saeed, and Moustafa Youssef. "RASID: A robust WLAN device-free passive motion detection system". In: *International Conference on Pervasive Computing and Communications* (Lugano, CHE, Mar. 19–23, 2012). PerCom. IEEE, Mar. 2012, pp. 180–189. DOI: 10.1109/PerCom.2012.6199865.

[Len19] Russell Lenth. *emmeans: Estimated Marginal Means, aka Least-Squares Means*. 2019. URL: https://cran.r-project.org/package=emmeans.

[Les+05] Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. "A Hybrid Discriminative/Generative

Approach for Modeling Human Activities". In: *International Joint Conference on Artificial Intelligence* (Edinburgh, GBR, July 30–Aug. 5, 2005). IJCAI. 2005.

[Li+17] Shengjie Li, Xiang Li, Kai Niu, Hao Wang, Yue Zhang, and Daqing Zhang. "AR-Alarm: An Adaptive and Robust Intrusion Detection System Leveraging CSI from Commodity Wi-Fi". In: *International Conference on Smart Homes and Health Telematics*. Enhanced Quality of Life and Smart Living (Paris, FRA, Aug. 29–31, 2017). ICOST. Springer International Publishing, 2017. ISBN: 3-319-66188-4. DOI: 10.1007/978-3-319-66188-9_18.

[LL13] Oscar D. Lara and Miguel A. Labrador. "A Survey on Human Activity Recognition using Wearable Sensors". In: *Communications Surveys & Tutorials* 15.3 (2013). ISSN: 1553-877X. DOI: 10.1109/SURV.2012.110112.00192.

[Lui+11] Gough Lui, Thomas Gallagher, Binghao Li, Andrew G. Dempster, and Chris Rizos. "Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization". In: *International Conference on Localization and GNSS*. ICL-GNSS. June 2011, pp. 53–57. DOI: 10.1109/ICL-GNSS.2011.5955283.

[Lv+18] Jiguang Lv, Dapeng Man, Wu Yang, Xiaojiang Du, and Miao Yu. "Robust WLAN-Based Indoor Intrusion Detection Using PHY Layer Information". In: *Access* 6 (2018). ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2785444.

[Mat+04] Merryn J. Mathie, Branko G. Celler, Nigel H. Lovell, and Adelle C. F. Coster. "Classification of basic daily movements using a triaxial accelerometer". In: *Medical and Biological Engineering and Computing* 42.5 (Sept. 2004), pp. 679–687. DOI: 10.1007/BF02347551.

[McK10] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Python in Science Conference*. Ed. by Stéfan van

der Walt and Jarrod Millman. 2010, pp. 51–56. DOI: 10.25080/MAJORA-92BF1922-00A.

[MY09] May Moussa and Moustafa Youssef. "Smart devices for smart environments: Device-free passive detection in real environments". In: *International Conference on Pervasive Computing and Communications* (Galveston, USA, Mar. 9–13, 2009). PerCom. IEEE, Mar. 2009. DOI: 10.1109/PERCOM.2009.4912826.

[MZW19] Yongsen Ma, Gang Zhou, and Shuangquan Wang. "WiFi Sensing with Channel State Information: A Survey". In: *ACM Computing Surveys* 52.3 (June 2019). ISSN: 0360-0300. DOI: 10.1145/3310194.

[ÖB14] Ahmet Turan Özdemir and Billur Barshan. "Detecting Falls with Wearable Sensors Using Machine Learning Techniques". In: *Sensors* 14.6 (June 2014). DOI: 10.3390/s140610691.

[OR16] Francisco Javier Ordóñez and Daniel Roggen. "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition". In: *Sensors* 16.1 (2016). ISSN: 1424-8220. DOI: 10.3390/s16010115.

[PAP16] Sameera Palipana, Piyush Agrawal, and Dirk Pesch. "Channel State Information Based Human Presence Detection Using Non-linear Techniques". In: *International Conference on Systems for Energy-Efficient Built Environments* (Palo Alto, CA, USA). BuildSys. ACM, 2016. DOI: 10.1145/2993422.2993579.

[Par12] Sang-Hyeun Park. "Efficient Decomposition-Based Multiclass and Multilabel Classification". MA thesis. Technische Universität Darmstadt, 2012.

[Pat+14] Neal Patwari, Joey Wilson, Sai Ananthanarayanan, Sneha K. Kasera, and Dwayne R. Westenskow. "Monitoring Breathing via Signal Strength in Wireless Networks". In: *Transactions on Mobile Computing* 13.8 (Aug. 2014). DOI: 10.1109/TMC.2013.117.

[Ped+11] Fabian Pedregosa, Gaël Varoquaux, Alexandr Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011).

[Pla99] John C. Platt. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". In: *Advances in Large Margin Classifiers*. MIT Press, 1999.

[Pom+02] Scott L. Pomeroy, Pablo Tamayo, Michelle Gaasenbeek, Lisa M. Sturla, Michael Angelo, Margaret E. McLaughlin, John Y. H. Kim, Liliana C. Goumnerova, Peter M. Black, Ching Lau, Jeffrey C. Allen, David Zagzag, James M. Olson, Tom Curran, Cynthia Wetmore, Jaclyn A. Biegel, Tomaso Poggio, Shayan Mukherjee, Ryan Rifkin, Andrea Califano, Gustavo Stolovitzky, David N. Louis, Jill P. Mesirov, Eric S. Lander, and Todd R. Golub. "Prediction of central nervous system embryonal tumour outcome based on gene expression". In: *Nature* 415.6870 (Jan. 24, 2002), pp. 436–442. DOI: 10.1038/415436a.

[PRV06] Oriol Pujol, Petia Radeva, and Jordi Vitrià. "Discriminant ECOC: A Heuristic Method for Application Dependent Design of Error Correcting Output Codes". In: *Transactions on Pattern Analysis and Machine Intelligence* 28.6 (June 2006), pp. 1007–1012. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2006.116.

[Qia+14] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Zimu Zhou. "PADS: Passive detection of moving targets with dynamic speed using PHY layer information". In: *International Conference on Parallel and Distributed Systems* (Hsinchu, TWN, Dec. 16–19, 2014). ICPADS. IEEE, Dec. 2014, pp. 1–8. DOI: 10.1109/PADSW.2014.7097784.

[Qia+18] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, Fugui He, and Tianzhang Xing. "Enabling Contactless Detection of Moving Humans with Dynamic Speeds Using CSI". In: *Transactions on Embedded Computing Systems* 17.2 (Jan. 2018). ISSN: 1539-9087. DOI: 10.1145/3157677.

[Raz+17] Abdul Rafiez Abdul Raziff, Md Nasir Sulaiman, Norwati Mustapha, and Thinagaran Perumal. "Single classifier, OvO, OvA and RCC multiclass classification method in handheld based smartphone gait identification". In: *International Conference on Applied Science and Technology* (Kedah, MYS, Apr. 3–5, 2017). ICAST. AIP Publishing, 2017. DOI: 10.1063/1.5005342.

[RCT19] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2019. URL: https://www.R-project.org/.

[RGM10] Juan J. Rodríguez, César García-Osorio, and Jesús Maudes. "Forests of nested dichotomies". In: *Pattern Recognition Letters* 31.2 (2010). ISSN: 0167-8655. DOI: 10.1016/j.patrec.2009.09.015.

[RS12] Attila Reiss and Didier Stricker. "Introducing a New Benchmarked Dataset for Activity Monitoring". In: *International Symposium on Wearable Computers* (Newcastle, GBR, June 18–22, 2012). ISWC. IEEE, June 2012. DOI: 10.1109/ISWC.2012.13.

[Rua+14] Wenjie Ruan, Lina Yao, Quan Z. Sheng, Nickolas J. G. Falkner, and Xue Li. "TagTrack: Device-free Localization and Tracking Using Passive RFID Tags". In: *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services* (London, GBR, Dec. 2–4, 2014). MOBIQUITOUS. Brussels, BEL: ICST, 2014. ISBN: 1-63190-039-0. DOI: 10.4108/icst.mobiquitous.2014.258004.

[San+17] Sadiq Sani, Nirmalie Wiratunga, Stewart Massie, and Kay Cooper. "kNN Sampling for Personalised Human Activity Recognition".

In: *International Conference on Case-Based Reasoning*. ICCBR. 2017. DOI: 10.1007/978-3-319-61030-6_23.

[San+18] Sadiq Sani, Nirmalie Wiratunga, Stewart Massie, and Kay Cooper. "Personalised Human Activity Recognition Using Matching Networks". In: *International Conference on Case-Based Reasoning* (Stockholm, SWE, July 9–12, 2018). ICCBR. 2018. DOI: 10.1007/978-3-030-01081-2_23.

[SCB20] Sebastian Scheurer, Tim Creedon, and Kenneth N. Brown. *A Wi-Fi Channel State Information (CSI) and Received Signal Strength (RSS) Data-set for Human Presence and Movement Detection*. 2020. DOI: 10.5281/zenodo.3653219.

[Sch+17a] Sebastian Scheurer, Salvatore Tedesco, Kenneth N. Brown, and Brendan O'Flynn. "Human Activity Recognition for Emergency First Responders via Body-Worn Inertial Sensors". In: *International Conference on Wearable and Implantable Body Sensor Networks* (Eindhoven, NLD, May 9–12, 2017). BSN. IEEE, May 2017. DOI: 10.1109/BSN.2017.7935994.

[Sch+17b] Sebastian Scheurer, Salvatore Tedesco, Kenneth N. Brown, and Brendan O'Flynn. "Sensor and Feature Selection for An Emergency First Responders Activity Recognition System". In: *Sensors* (Glasgow, GBR, Oct. 29–Nov. 1, 2017). IEEE, Oct. 2017. DOI: 10.1109/ICSENS.2017.8234090.

[Sch+18] Sebastian Scheurer, Salvatore Tedesco, Òscar Manzano, Kenneth N. Brown, and Brendan O'Flynn. "Monitoring Emergency First Responders' Activities via Gradient Boosting and Inertial Sensor Data". In: *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases* (Dublin, IRL, Sept. 10–14, 2018). ECML/PKDD. 2018. DOI: 10.1007/978-3-030-10997-4_53.

[Sch+19] Sebastian Scheurer, Salvatore Tedesco, Kenneth N. Brown, and Brendan O'Flynn. "Subject-Dependent and -Independent Hu-

man Activity Recognition with Person-Specific and -Independent Models". In: *International Workshop on Sensor-based Activity Recognition and Interaction* (Rostock, DEU, Sept. 16–17, 2019). iWOAR. ACM, Sept. 2019. DOI: 10.1145/3361684.3361689.

[Sch+20a] Sebastian Scheurer, Salvatore Tedesco, Kenneth N. Brown, and Brendan O'Flynn. "Using Domain Knowledge for Interpretable and Competitive Multi-class Human Activity Recognition". In: *Sensors* 20.4 (Feb. 2020): *Inertial Sensors for Activity Recognition and Classification*. ISSN: 1424-8220. DOI: 10.3390/s20041208.

[Sch+20b] Sebastian Scheurer, Salvatore Tedesco, Brendan O'Flynn, and Kenneth N. Brown. "Comparing Person-Specific and -Independent Models on Subject-Dependent and -Independent Human Activity Recognition Performance". In: *Sensors* 20.13 (June 2020): *Sensor-Based Activity Recognition and Interaction*. ISSN: 1424-8220. DOI: 10.3390/s20133647.

[SF11] Carlos N. Silla and Alex A. Freitas. "A survey of hierarchical classification across different application domains". In: *Data Mining and Knowledge Discovery* 22.1 (Jan. 2011), pp. 31–72. ISSN: 1573-756X. DOI: 10.1007/s10618-010-0175-9.

[SFR18] Daniel Silva-Palacios, Cèsar Ferri, and María José Ramírez-Quintana. "Probabilistic class hierarchies for multiclass classification". In: *Journal of Computational Science* 26 (May 2018), pp. 254–263. ISSN: 1877-7503. DOI: 10.1016/j.jocs.2018.01.006.

[Sho+14] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul J. M. Havinga. "Fusion of Smartphone Motion Sensors for Physical Activity Recognition". In: *Sensors* 14.6 (2014). DOI: 10.3390/s140610146.

[Sho+16] Muhammad Shoaib, Hans Scholten, Paul J. M. Havinga, and Ozlem Durmaz Incel. "A hierarchical lazy smoking detection algorithm using smartwatch sensors". In: *International Conference on*

e-Health Networking, Applications and Services (Munich, DEU, Sept. 14–16, 2016). IEEE, Sept. 2016. DOI: 10.1109/HealthCom.2016.7749439.

[Sig+13] Stephan Sigg, Shuyu Shi, Felix Buesching, Yusheng Ji, and Lars Wolf. "Leveraging RF-channel fluctuation for activity recognition. Active and passive systems, continuous and RSSI-based signal features". In: *International Conference on Advances in Mobile Computing & Multimedia*. MoMM. Dec. 2013, pp. 43–52.

[Sig+14] Stephan Sigg, Markus Scholz, Shuyu Shi, Yusheng Ji, and Michael Beigl. "RF-Sensing of Activities from Non-Cooperative Subjects in Device-Free Recognition Systems Using Ambient and Local Signals". In: *Transactions on Mobile Computing* 13.4 (Apr. 2014). ISSN: 1536-1233. DOI: 10.1109/TMC.2013.28.

[SS16] Timo Sztyler and Heiner Stuckenschmidt. "On-body localization of wearable devices: An investigation of position-aware activity recognition". In: *International Conference on Pervasive Computing and Communications* (Sydney, AUS, Mar. 14–19, 2016). PerCom. IEEE, Mar. 2016. DOI: 10.1109/PERCOM.2016.7456521.

[Tan11] Ole Tange. "GNU Parallel—The Command-Line Power Tool". In: *;login: The USENIX Magazine* 36.1 (Feb. 2011). URL: http://www.gnu.org/s/parallel.

[Ted+20] Salvatore Tedesco, Colum Crowe, Andrew Ryan, Marco Sica, Sebastian Scheurer, Amanda Clifford, Kenneth N. Brown, and Brendan O'Flynn. "Motion Sensors-based Machine Learning Approach for the Identification of Anterior Cruciate Ligament Gait Patterns in On-the-Field Activities in Rugby Players". In: *Sensors* 20.11 (May 2020). ISSN: 1424-8220. DOI: 10.3390/s20113029.

[TKO15] Salvatore Tedesco, Jasurbek Khodjaev, and Brendan O'Flynn. "A novel first responders location tracking system: Architecture and functional requirements". In: *Mediterranean Microwave Sym-*

posium (Lecce, ITA). MMS. IEEE. Nov. 2015. DOI: 10.1109/MMS.2015.7375416.

[VGR20] Meysam Vakili, Mohammad Ghamsari, and Masoumeh Rezaei. "Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification". In: *CoRR* (Jan. 2020). arXiv: 2001.09636 [cs.LG].

[Wan+19] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. "Deep learning for sensor-based activity recognition: A survey". In: *Pattern Recognition Letters* 119 (2019), pp. 3–11. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2018.02.010.

[WCV11] Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation". In: *Computing in Science Engineering* 13.2 (Mar. 2011), pp. 22–30. ISSN: 1521-9615. DOI: 10.1109/MCSE.2011.37.

[WL12] Gary Mitchell Weiss and Jeffrey W. Lockhart. "The Impact of Personalization on Smartphone-Based Activity Recognition". In: *Conference on Artificial Intelligence*. Workshop on Activity Context Representation: Techniques and Languages. AAAI, 2012.

[WPH06] Kristen Woyach, Daniele Puccinelli, and Martin Haenggi. "Sensorless sensing in wireless networks: Implementation and measurements". In: *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks* (Vienna, AUT). IEEE. ACM, 2006. DOI: 10.1145/2536853.2536873.

[Wu+15] Chenshu Wu, Zheng Yang, Zimu Zhou, Xuefeng Liu, Yunhao Liu, and Jiannong Cao. "Non-Invasive Detection of Moving and Stationary Human With WiFi". In: *Journal on Selected Areas in Communications* 33.11 (Nov. 2015). DOI: 10.1109/JSAC.2015.2430294.

[WW98] Jason Weston and Chris Watkins. *Multi-class Support Vector Machines*. Tech. rep. Royal Holloway University of London, May 1998.

[Xia+12] Jiang Xiao, Kaishun Wu, Youwen Yi, Lu Wang, and Lionel M. Ni. "FIMD: Fine-grained Device-free Motion Detection". In: *International Conference on Parallel and Distributed Systems*. IEEE, Dec. 2012, pp. 229–235. DOI: 10.1109/ICPADS.2012.40.

[XLL15] Yaxiong Xie, Zhenjiang Li, and Mo Li. "Precise Power Delay Profiling with Commodity WiFi". In: *Annual International Conference on Mobile Computing and Networking* (Paris, FRA). MobiCom. New York, NY, USA: ACM, 2015, pp. 53–64. ISBN: 978-1-4503-3619-2. DOI: 10.1145/2789168.2790124.

[Xu+13] Chenren Xu, Bernhard Firner, Robert S. Moore, Yanyong Zhang, Wade Trappe, Richard Howard, Feixiong Zhang, and Ning An. "SCPL: Indoor device-free multi-subject counting and localization using radio signal strength". In: *International Conference on Information Processing in Sensor Networks*. IPSN. IEEE, Apr. 2013. DOI: 10.1145/2461381.2461394.

[Yan+10] Jie Yang, Yong Ge, Hui Xiong, Yingying Chen, and Hongbo Liu. "Performing Joint Learning for Passive Intrusion Detection in Pervasive Wireless Environments". In: *Conference on Computer Communications* (San Diego, USA, Mar. 14–19, 2010). INFOCOM. IEEE, Mar. 2010. DOI: 10.1109/INFCOM.2010.5462148.

[Yea+01] Chen-Hsiang Yeang, Sridhar Ramaswamy, Pablo Tamayo, Sayan Mukherjee, Ryan M. Rifkin, Michael Angelo, Michael Reich, Eric Lander, Jill Mesirov, and Todd Golub. "Molecular classification of multiple tumor types". In: *Bioinformatics* 17.suppl. 1 (June 1, 2001). DOI: 10.1093/bioinformatics/17.suppl_1.S316.

[YJ16] Hüseyin Yiğitler and Riku Jäntti. "Experimental accuracy assessment of radio tomographic imaging methods". In: *International Conference on Pervasive Computing and Communications* (Sydney, AUS, Mar. 14–18, 2016). PerCom. IEEE, Mar. 2016. DOI: 10.1109/PERCOMW.2016.7457117.

[YMA07] Moustafa Youssef, Matthew Mah, and Ashok Agrawala. "Challenges: Device-free Passive Localization for Wireless Environments". In: *International Conference on Mobile Computing and Networking* (Montréal, CAN). MobiCom. New York, NY, USA: ACM, 2007. DOI: 10.1145/1287853.1287880.

[YZL13] Zheng Yang, Zimu Zhou, and Yunhao Liu. "From RSSI to CSI. Indoor Localization via Channel Response". In: *ACM Computing Surveys* 46.2 (Dec. 2013). DOI: 10.1145/2543581.2543592.

[ZAK16] Mingmin Zhao, Fadel Adib, and Dina Katabi. "Emotion Recognition Using Wireless Signals". In: *Annual International Conference on Mobile Computing and Networking* (New York, USA, Oct. 3–7, 2016). MobiCom. New York, NY, USA: ACM, Oct. 2016. ISBN: 1-4503-4226-4. DOI: 10.1145/2973750.2973762.

[Zha+16] Xiao Zhang, Jie Wang, Qinghua Gao, Xiaorui Ma, and Hongyu Wang. "Device-free wireless localization and activity recognition with deep learning". In: *International Conference on Pervasive Computing and Communications* (Sydney, AUS, Mar. 14–18, 2016). PerCom. IEEE. Mar. 2016. DOI: 10.1109/PERCOMW.2016.7457118.

[Zha+18] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. "Through-Wall Human Pose Estimation Using Radio Signals". In: *Conference on Computer Vision and Pattern Recognition* (Salt Lake City, USA, June 18–23, 2018). CVPR. IEEE, June 2018. DOI: 10.1109/CVPR.2018.00768.

[Zhe+19] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. "Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi". In: *Annual International Conference on Mobile Systems, Applications, and Services* (Seoul, KOR). MobiSys. New York, NY, USA: ACM, June 2019, pp. 313–325. ISBN: 9781450366618. DOI: 10.1145/3307334.3326081.

[Zho+13] Zimu Zhou, Zheng Yang, Chenshu Wu, Longfei Shangguan, and Yunhao Liu. "Towards omnidirectional passive human detection". In: *Conference on Computer Communications* (Turin, ITA, Apr. 14–19, 2013). INFOCOM. IEEE, Apr. 2013, pp. 3057–3065. DOI: 10.1109/INFCOM.2013.6567118.

[Zho+14] Zimu Zhou, Zheng Yang, Chenshu Wu, Longfei Shangguan, and Yunhao Liu. "Omnidirectional Coverage for Device-Free Passive Human Detection". In: *Transactions on Parallel and Distributed Systems* 25.7 (July 2014). DOI: 10.1109/TPDS.2013.274.

[Zho+15] Zimu Zhou, Zheng Yang, Chenshu Wu, Yunhao Liu, and Lionel M. Ni. "On Multipath Link Characterization and Adaptation for Device-Free Human Detection". In: *International Conference on Distributed Computing Systems*. IEEE, June 2015, pp. 389–398. DOI: 10.1109/ICDCS.2015.47.

[Zho+19] Mu Zhou, Yaoping Li, Liangbo Xie, and Wei Nie. "Maximum Mean Discrepancy Minimization Based Transfer Learning for Indoor WLAN Personnel Intrusion Detection". In: *Sensors Letters* 3.8 (Aug. 2019). DOI: 10.1109/LSENS.2019.2932099.

[Zhu+17] Hai Zhu, Fu Xiao, Lijuan Sun, Ruchuan Wang, and Panlong Yang. "R-TTWD: Robust Device-Free Through-The-Wall Detection of Moving Human With WiFi". In: *Journal on Selected Areas in Communications* 35.5 (May 2017). DOI: 10.1109/JSAC.2017.2679578.

[Zim+10] Arthur Zimek, Fabian Buchwald, Eibe Frank, and Stefan Kramer. "A Study of Hierarchical and Flat Classification of Proteins". In: *Transactions on Computational Biology and Bioinformatics* 7.3 (July 2010). ISSN: 1545-5963. DOI: 10.1109/TCBB.2008.104.

[ZWL15] Licheng Zhang, Xihong Wu, and Dingsheng Luo. "Recognizing Human Activities from Raw Accelerometer Data Using Deep Neural Networks". In: *International Conference on Machine Learning*

and Applications (Miami, USA, Dec. 9–11, 2015). ICMLA. IEEE, Dec. 2015, pp. 865–870. DOI: 10.1109/ICMLA.2015.48.

Appendix A

Additional Performance Metrics for Micro and Macro Models

In this appendix, we provide alternative performance measures for the results presented in chapter 4. Table A.1 lists the mean subject-dependent and -independent accuracy score—in percent ± standard error (SE)—achieved with person-independent models (PIMs), person-specific models (PSMs), unweighted ensembles of person-specific models (EPSMs), κ-weighted ensembles of person-specific models (WEPSM$_\kappa$s), and baseline-feature-weighted ensembles of person-specific models (WEPSM$_{\text{bf}}$s), and Table A.2 the mean weighted F1-score for same. The metrics were chosen to match the ones from the literature discussed in subsection 2.1.1.

Table A.1: Subject-dependent and -independent accuracy (%) ± SE when machine learning algorithms (MLA) are combined with a PIM, a PSM, an unweighted EPSM, a WEPSM_κ, or a WEPSM_{bf}

		subject-dependent		subject-independent				
dataset	MLA	PIM	(E)PSM	PIM	PSM	EPSM	WEPSM_κ	WEPSM_{bf}
FUSION	gbt	98.3 ± 0.3	98.0 ± 0.3	93.5 ± 1.8	84.1 ± 1.8	91.9 ± 2.2	92.1 ± 2.2	91.7 ± 2.2
wrist	knn	94.9 ± 0.8	95.0 ± 0.9	87.9 ± 1.7	78.5 ± 2.2	89.3 ± 2.4	89.3 ± 2.4	89.1 ± 2.4
	glm	97.2 ± 0.5	97.8 ± 0.3	93.0 ± 1.8	82.3 ± 2.3	91.0 ± 2.4	91.0 ± 2.4	90.6 ± 2.3
	svm	98.2 ± 0.3	98.1 ± 0.3	92.2 ± 1.8	82.9 ± 2.0	91.7 ± 2.3	91.8 ± 2.3	91.4 ± 2.3
MHEALTH	gbt	97.8 ± 0.7	97.5 ± 1.1	84.0 ± 3.3	63.2 ± 2.1	74.7 ± 3.2	74.9 ± 3.1	74.2 ± 3.2
wrist	knn	93.6 ± 1.2	94.3 ± 1.3	78.3 ± 2.8	60.0 ± 1.9	74.0 ± 2.3	74.2 ± 2.4	75.3 ± 2.3
	glm	93.8 ± 1.2	96.1 ± 1.3	80.9 ± 3.0	58.3 ± 2.2	72.8 ± 2.6	72.9 ± 2.6	73.4 ± 2.6
	svm	95.9 ± 0.9	97.0 ± 0.9	83.6 ± 2.4	61.8 ± 1.9	74.5 ± 3.6	74.7 ± 3.6	74.1 ± 3.6
OPPORT	gbt	87.8 ± 1.9	89.1 ± 1.5	79.8 ± 4.6	72.4 ± 3.4	78.7 ± 4.9	78.5 ± 5.0	78.7 ± 4.8
wrist	knn	80.7 ± 1.8	83.2 ± 1.8	67.7 ± 2.7	61.6 ± 2.2	70.0 ± 3.4	69.5 ± 3.1	70.1 ± 2.8
	glm	81.5 ± 2.5	84.6 ± 1.9	73.8 ± 4.3	63.9 ± 2.1	72.0 ± 4.0	71.8 ± 4.2	72.2 ± 3.8
	svm	87.0 ± 1.7	87.4 ± 1.6	77.4 ± 4.3	65.6 ± 1.8	75.3 ± 4.0	75.1 ± 4.1	75.6 ± 3.7
PAMAP2	gbt	88.8 ± 0.4	88.9 ± 0.6	79.6 ± 3.9	59.2 ± 2.7	75.2 ± 3.8	75.6 ± 3.9	74.8 ± 3.5
chest	knn	78.0 ± 0.9	80.6 ± 1.1	67.3 ± 2.3	54.1 ± 1.7	71.0 ± 3.0	71.4 ± 3.1	69.8 ± 2.8
	glm	84.3 ± 1.0	86.9 ± 0.8	75.0 ± 3.4	53.9 ± 2.2	72.4 ± 4.4	72.4 ± 4.5	71.7 ± 4.4
	svm	87.4 ± 0.7	86.6 ± 0.7	76.3 ± 4.1	54.6 ± 2.5	72.4 ± 4.7	72.9 ± 4.7	71.6 ± 4.7

Continued on next page

Continued from previous page

		subject-dependent		subject-independent				
dataset	MLA	PIM	(E)PSM	PIM	PSM	EPSM	WEPSM_{κ}	WEPSM_{bf}
PAMAP2	gbt	88.1 ± 1.0	87.4 ± 0.8	80.6 ± 2.6	61.1 ± 2.2	74.6 ± 2.4	75.0 ± 2.4	74.8 ± 2.6
wrist	knn	79.7 ± 1.3	81.1 ± 1.4	68.7 ± 3.7	52.6 ± 2.5	71.4 ± 3.5	71.7 ± 3.6	70.9 ± 3.4
	glm	84.2 ± 1.6	85.2 ± 1.2	77.2 ± 3.7	54.8 ± 3.3	72.0 ± 4.5	72.4 ± 4.5	72.1 ± 4.4
	svm	86.5 ± 1.1	85.0 ± 1.1	75.6 ± 4.8	51.6 ± 2.8	71.5 ± 4.1	72.0 ± 4.1	71.3 ± 3.9
REALWORLD	gbt	94.5 ± 0.5	96.9 ± 0.3	76.4 ± 3.7	47.1 ± 1.7	68.7 ± 3.4	69.9 ± 3.3	69.2 ± 3.4
chest	knn	88.1 ± 1.1	92.9 ± 0.8	66.1 ± 2.9	47.2 ± 2.1	68.3 ± 3.2	69.2 ± 3.1	68.5 ± 3.3
	glm	86.8 ± 1.4	96.3 ± 0.4	66.4 ± 5.5	40.7 ± 2.0	64.0 ± 3.8	65.3 ± 3.8	64.0 ± 4.0
	svm	93.6 ± 0.6	96.4 ± 0.4	68.1 ± 4.6	40.6 ± 2.0	62.0 ± 4.2	63.3 ± 4.0	62.2 ± 4.3
SAFESENS	gbt	94.7 ± 0.7	97.3 ± 0.8	71.2 ± 2.6	33.3 ± 1.9	53.1 ± 4.5	52.7 ± 5.3	57.9 ± 3.1
chest	knn	83.0 ± 1.8	89.0 ± 1.4	59.9 ± 3.6	35.8 ± 1.8	58.9 ± 3.3	58.9 ± 3.4	58.0 ± 2.5
	glm	80.9 ± 1.6	93.7 ± 1.0	67.7 ± 2.8	33.1 ± 1.7	58.6 ± 2.5	58.0 ± 2.4	58.0 ± 2.6
	svm	89.2 ± 1.2	95.7 ± 0.8	70.2 ± 2.6	35.8 ± 1.8	56.7 ± 2.4	57.8 ± 2.3	56.1 ± 2.8
SIMFALL	gbt	60.0 ± 1.1	68.1 ± 1.2	47.5 ± 1.5	24.6 ± 0.6	37.8 ± 1.4	37.8 ± 1.4	37.0 ± 1.4
chest	knn	48.6 ± 0.9	52.8 ± 1.2	34.8 ± 0.7	25.0 ± 0.6	37.7 ± 1.0	37.5 ± 1.0	36.5 ± 0.9
	glm	42.3 ± 0.8	55.4 ± 1.1	38.8 ± 1.1	20.0 ± 0.4	33.6 ± 0.9	33.6 ± 0.9	32.7 ± 0.9
	svm	53.4 ± 0.7	52.8 ± 1.5	42.2 ± 1.3	19.7 ± 0.4	30.7 ± 0.9	31.0 ± 0.9	29.6 ± 0.9
SIMFALL	gbt	58.3 ± 1.4	65.1 ± 1.4	44.6 ± 2.2	24.4 ± 0.9	37.3 ± 2.0	37.4 ± 2.0	36.4 ± 2.1
wrist	knn	48.2 ± 1.1	52.1 ± 1.1	33.8 ± 1.4	24.4 ± 0.9	36.2 ± 1.5	36.3 ± 1.5	35.4 ± 1.6
	glm	41.3 ± 1.3	52.9 ± 1.2	37.0 ± 1.9	21.5 ± 0.8	32.3 ± 1.6	32.8 ± 1.6	32.1 ± 1.7

Continued on next page

Continued from previous page

		subject-dependent		subject-independent				
dataset	MLA	PIM	(E)PSM	PIM	PSM	EPSM	WEPSM_{κ}	WEPSM_{bf}
	svm	51.5 ± 1.2	49.4 ± 1.4	40.1 ± 2.2	19.2 ± 0.7	31.6 ± 1.4	31.9 ± 1.5	31.1 ± 1.6
UTSMOKE	gbt	83.6 ± 1.3	92.1 ± 0.8	73.2 ± 2.5	61.3 ± 1.6	70.3 ± 2.8	70.4 ± 2.8	70.2 ± 2.7
wrist	knn	79.7 ± 1.1	83.9 ± 1.0	67.1 ± 2.1	57.8 ± 1.5	66.3 ± 2.4	66.4 ± 2.5	66.1 ± 2.3
	glm	73.3 ± 1.8	86.4 ± 1.0	68.5 ± 2.2	57.6 ± 1.4	65.1 ± 2.1	65.2 ± 2.2	65.3 ± 2.0
	svm	86.0 ± 1.1	90.7 ± 0.8	73.6 ± 2.3	59.6 ± 1.6	68.8 ± 2.5	68.9 ± 2.5	69.0 ± 2.3

Table A.2: Subject-dependent and -independent weighted F1-score (%) ± SE when machine learning algorithms (MLA) are combined with a PIM, a PSM, an unweighted EPSM, a WEPSM_κ, or a WEPSM_{bf}

		subject-dependent		subject-independent				
dataset	MLA	PIM	(E)PSM	PIM	PSM	EPSM	WEPSM_κ	WEPSM_{bf}
FUSION	gbt	98.2 ± 0.3	98.0 ± 0.3	92.7 ± 2.3	82.3 ± 2.1	90.6 ± 2.8	90.8 ± 2.8	90.4 ± 2.8
wrist	knn	94.9 ± 0.8	95.0 ± 0.9	87.7 ± 1.8	77.4 ± 2.5	88.2 ± 2.9	88.4 ± 2.9	88.2 ± 2.9
	glm	97.2 ± 0.5	97.8 ± 0.3	92.5 ± 2.2	80.7 ± 2.5	89.8 ± 2.9	89.9 ± 2.9	89.4 ± 2.9
	svm	98.2 ± 0.3	98.1 ± 0.3	91.6 ± 2.1	81.2 ± 2.3	90.4 ± 2.9	90.5 ± 3.0	90.1 ± 2.9
MHEALTH	gbt	97.7 ± 0.7	97.5 ± 1.1	81.9 ± 4.0	58.3 ± 2.2	69.8 ± 3.6	70.1 ± 3.5	69.2 ± 3.6
wrist	knn	93.5 ± 1.2	94.1 ± 1.3	76.6 ± 2.8	56.3 ± 1.9	70.9 ± 2.7	71.2 ± 2.7	72.6 ± 2.6
	glm	93.7 ± 1.3	96.1 ± 1.3	78.9 ± 3.3	53.0 ± 2.1	67.6 ± 3.2	67.8 ± 3.2	68.2 ± 3.2
	svm	95.9 ± 0.9	97.0 ± 0.9	81.9 ± 2.7	56.6 ± 2.0	70.0 ± 4.1	70.2 ± 4.1	69.3 ± 4.1
OPPORT	gbt	87.8 ± 1.9	89.1 ± 1.5	79.3 ± 5.0	71.4 ± 4.2	77.3 ± 5.7	77.0 ± 5.8	77.4 ± 5.5
wrist	knn	80.7 ± 1.8	83.2 ± 1.8	67.2 ± 2.9	61.1 ± 2.3	69.2 ± 3.6	68.7 ± 3.4	69.5 ± 3.0
	glm	81.3 ± 2.4	84.5 ± 1.9	72.7 ± 4.9	63.0 ± 2.6	70.3 ± 4.7	70.2 ± 4.9	70.8 ± 4.5
	svm	87.0 ± 1.8	87.4 ± 1.6	76.6 ± 4.8	64.5 ± 2.2	73.9 ± 4.7	73.6 ± 5.0	74.4 ± 4.4
PAMAP2	gbt	89.0 ± 0.4	89.1 ± 0.6	79.2 ± 4.6	56.2 ± 3.3	74.3 ± 4.5	74.8 ± 4.6	73.9 ± 4.2
chest	knn	78.2 ± 1.0	80.6 ± 1.1	67.7 ± 2.5	52.4 ± 2.1	70.9 ± 3.4	71.3 ± 3.5	69.5 ± 3.2
	glm	84.3 ± 1.1	87.0 ± 0.8	74.1 ± 4.1	50.4 ± 2.7	71.4 ± 5.1	71.4 ± 5.2	70.4 ± 5.1
	svm	87.6 ± 0.7	86.6 ± 0.7	75.9 ± 4.6	51.5 ± 3.0	71.2 ± 5.6	71.7 ± 5.5	70.3 ± 5.5

Continued on next page

Continued from previous page

		subject-dependent		subject-independent				
dataset	MLA	PIM	(E)PSM	PIM	PSM	EPSM	WEPSM_{κ}	WEPSM_{bf}
PAMAP2	gbt	88.3 ± 1.0	87.6 ± 0.8	79.9 ± 3.0	58.2 ± 2.4	73.1 ± 2.6	73.5 ± 2.6	73.3 ± 2.8
wrist	knn	79.7 ± 1.3	81.0 ± 1.4	68.2 ± 4.1	50.5 ± 2.8	70.3 ± 3.9	70.7 ± 3.9	69.5 ± 3.8
	glm	84.2 ± 1.6	85.3 ± 1.2	76.5 ± 4.2	51.5 ± 3.7	70.0 ± 5.3	70.6 ± 5.2	70.1 ± 5.2
	svm	86.6 ± 1.1	85.1 ± 1.1	75.0 ± 5.3	48.3 ± 3.3	69.7 ± 4.8	70.3 ± 4.8	69.4 ± 4.7
REALWORLD	gbt	94.7 ± 0.5	96.8 ± 0.3	76.5 ± 3.6	43.4 ± 1.6	68.4 ± 3.4	70.0 ± 3.2	68.7 ± 3.2
chest	knn	88.8 ± 1.0	92.9 ± 0.8	67.5 ± 2.9	45.3 ± 2.1	67.9 ± 3.5	69.0 ± 3.3	68.4 ± 3.4
	glm	87.5 ± 1.4	96.2 ± 0.4	66.3 ± 5.6	37.2 ± 2.2	62.0 ± 3.9	63.8 ± 3.9	62.1 ± 3.9
	svm	93.9 ± 0.5	96.3 ± 0.4	67.8 ± 4.9	36.9 ± 2.0	59.4 ± 4.4	61.5 ± 4.2	59.8 ± 4.4
SAFESENS	gbt	95.1 ± 0.7	97.3 ± 0.8	70.8 ± 2.6	29.0 ± 1.7	52.5 ± 3.9	51.3 ± 4.8	55.1 ± 3.6
chest	knn	83.5 ± 1.9	88.9 ± 1.4	61.0 ± 4.0	33.3 ± 2.2	59.7 ± 3.9	59.6 ± 4.0	57.1 ± 3.1
	glm	81.4 ± 1.8	93.6 ± 1.0	67.7 ± 3.0	29.9 ± 2.1	58.1 ± 2.8	57.6 ± 2.9	56.3 ± 3.1
	svm	89.6 ± 1.2	95.6 ± 0.8	70.5 ± 3.0	31.8 ± 2.2	56.8 ± 2.9	57.7 ± 3.0	54.0 ± 3.3
SIMFALL	gbt	59.8 ± 1.1	68.3 ± 1.2	46.9 ± 1.5	22.4 ± 0.6	34.7 ± 1.5	34.9 ± 1.5	33.9 ± 1.5
chest	knn	48.7 ± 0.9	52.9 ± 1.2	34.6 ± 0.7	23.7 ± 0.6	36.0 ± 1.0	36.1 ± 1.1	34.9 ± 0.9
	glm	41.9 ± 0.9	55.3 ± 1.2	38.2 ± 1.1	17.7 ± 0.5	31.0 ± 1.0	31.1 ± 1.0	30.0 ± 1.0
	svm	53.6 ± 0.7	52.7 ± 1.5	41.8 ± 1.3	17.4 ± 0.4	27.8 ± 1.1	28.3 ± 1.0	26.4 ± 1.1
SIMFALL	gbt	58.2 ± 1.4	65.2 ± 1.4	44.2 ± 2.1	22.1 ± 0.9	35.7 ± 1.9	35.9 ± 1.9	34.7 ± 1.9
wrist	knn	48.2 ± 1.1	52.2 ± 1.1	33.5 ± 1.4	23.5 ± 0.9	35.5 ± 1.5	35.7 ± 1.5	34.5 ± 1.6
	glm	41.0 ± 1.3	52.8 ± 1.2	36.5 ± 1.9	19.6 ± 0.8	30.8 ± 1.5	31.3 ± 1.6	29.9 ± 1.6

Continued on next page

Continued from previous page

		subject-dependent		subject-independent				
dataset	MLA	PIM	(E)PSM	PIM	PSM	EPSM	WEPSM_{κ}	WEPSM_{bf}
	svm	51.8 ± 1.2	49.4 ± 1.4	39.8 ± 2.2	17.3 ± 0.6	29.1 ± 1.2	29.5 ± 1.3	28.2 ± 1.4
UTSMOKE	gbt	83.3 ± 1.3	92.2 ± 0.8	72.3 ± 2.6	59.2 ± 1.6	69.1 ± 2.8	69.1 ± 2.8	69.0 ± 2.7
wrist	knn	79.5 ± 1.2	83.8 ± 1.0	66.6 ± 2.1	55.6 ± 1.5	64.0 ± 2.5	64.1 ± 2.5	63.7 ± 2.4
	glm	72.4 ± 1.9	86.2 ± 1.1	67.2 ± 2.2	54.7 ± 1.4	62.7 ± 2.2	62.8 ± 2.2	63.0 ± 2.2
	svm	85.9 ± 1.1	90.7 ± 0.8	73.0 ± 2.3	57.1 ± 1.6	67.0 ± 2.5	67.1 ± 2.6	67.2 ± 2.4

Appendix B

Expert Hierarchies

In this appendix, we show the expert hierarchies that are the topic of chapter 5. Figures B.1–B.5 illustrate the expert hierarchies 1 through 5.

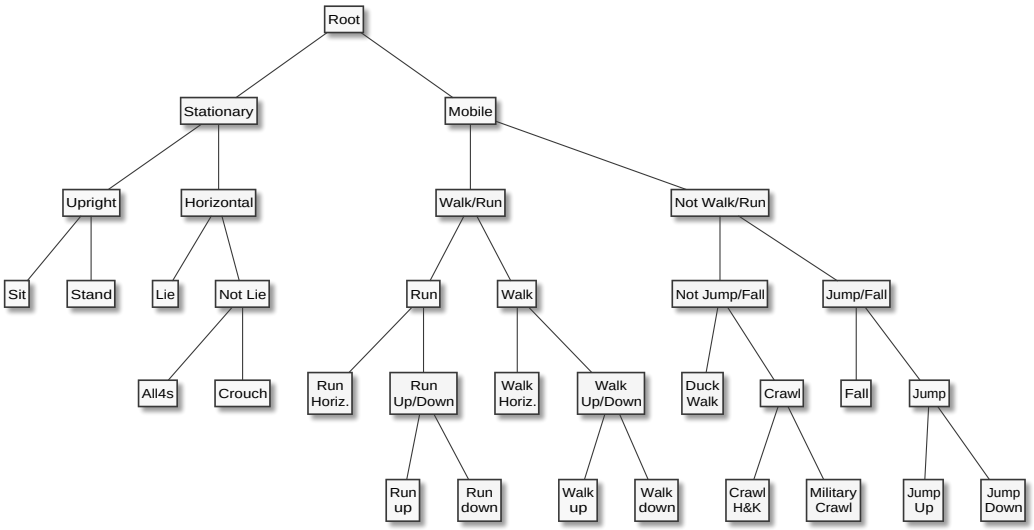


Figure B.1: Expert hierarchy 1 (EH1)

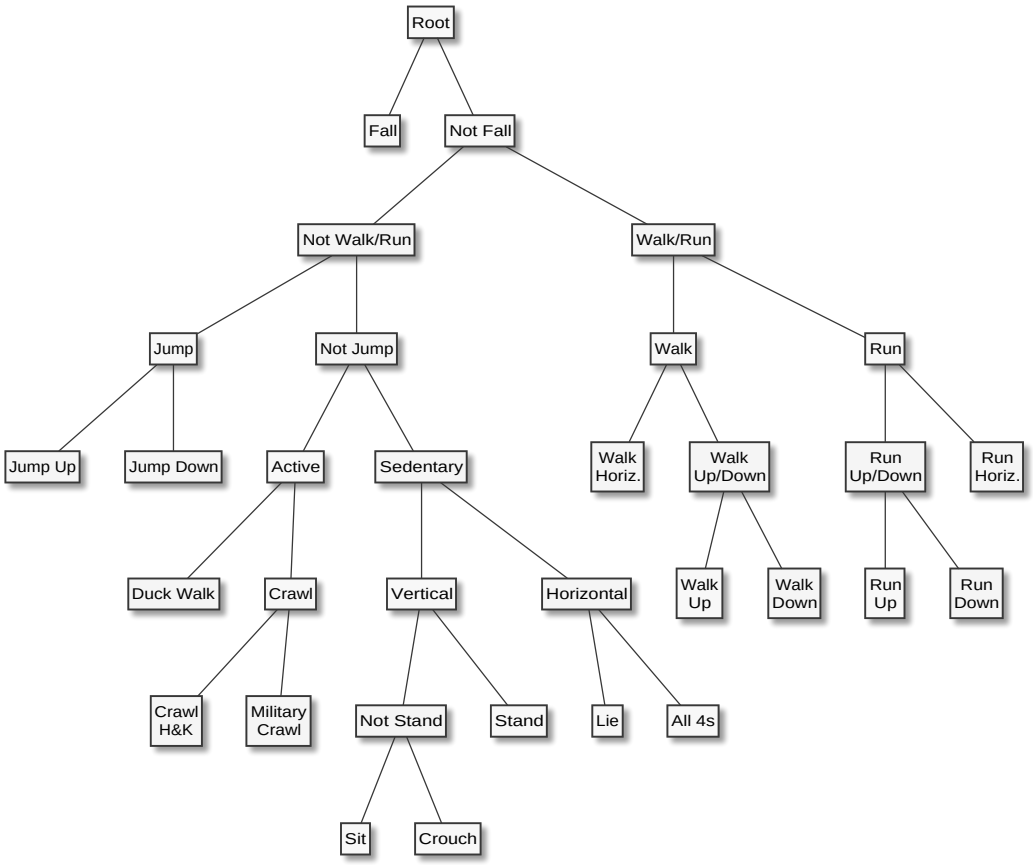


Figure B.2: Expert hierarchy 2 (EH2)

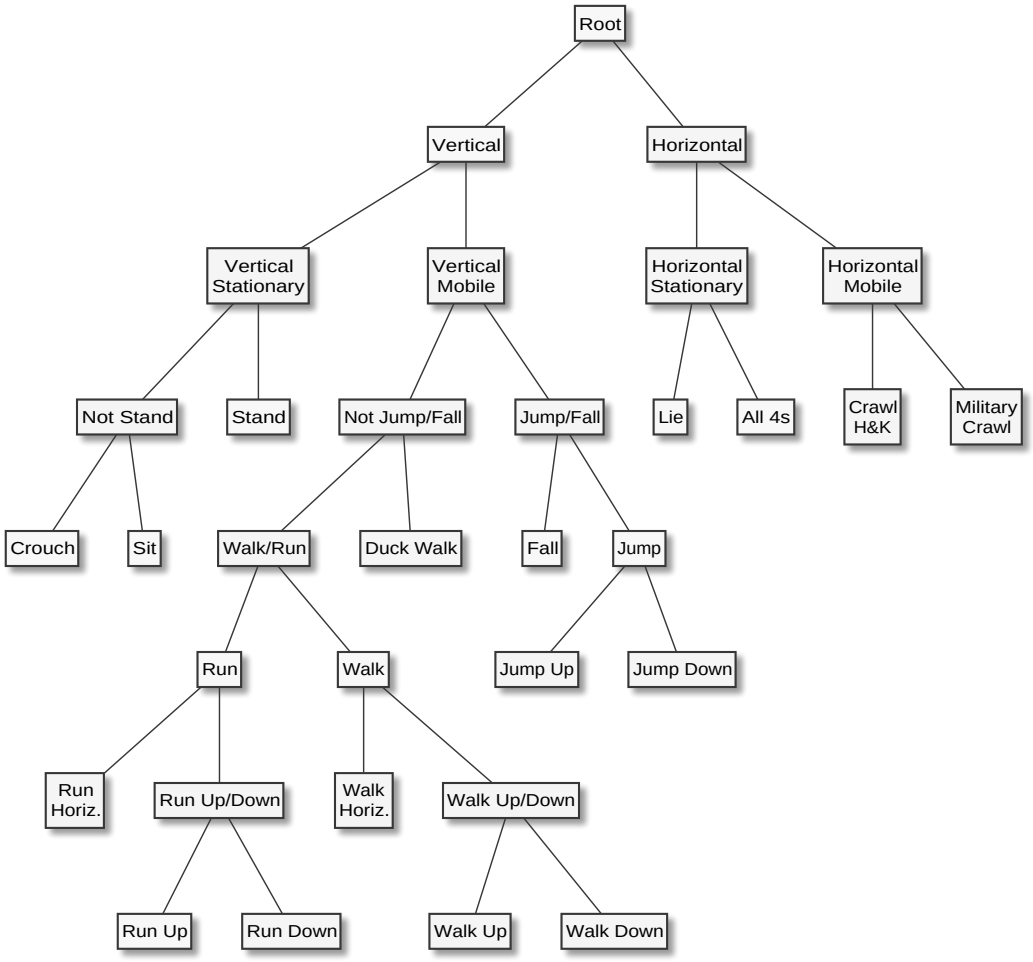


Figure B.3: Expert hierarchy 3 (EH3)

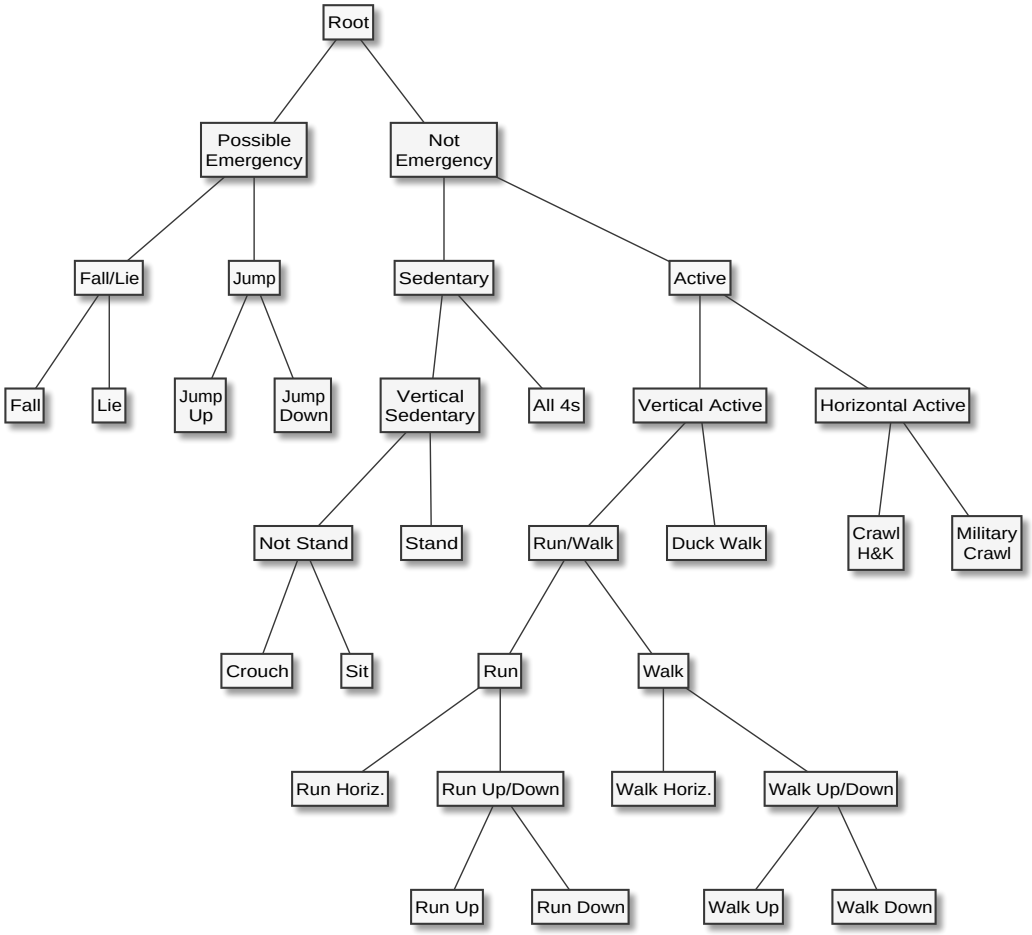


Figure B.4: Expert hierarchy 4 (EH4)

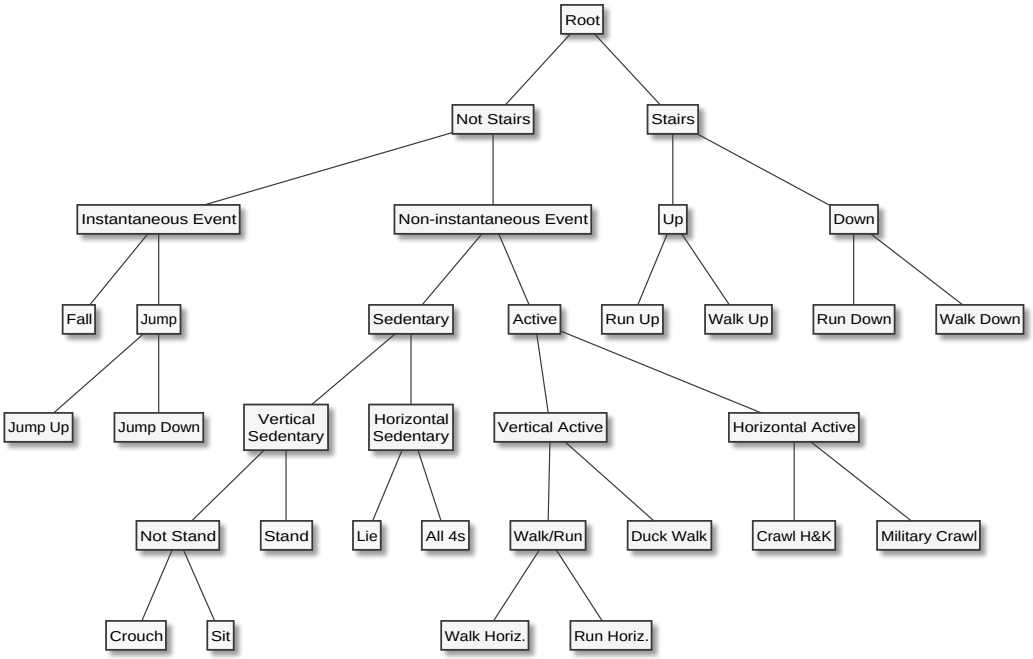


Figure B.5: Expert hierarchy 5 (EH5)