Technical University of Kaiserslautern

Personalized Mobile Physical Activity Monitoring for Everyday Life

by

Attila Reiss

Thesis approved by the
Department of Computer Science
Technical University of Kaiserslautern
for the award of the doctoral degree:

Doctor of Engineering (Dr.-Ing.)

Dean: Prof. Dr. Arnd Poetzsch-Heffter
Chair of the committee: Prof. Dr. Paul Müller
Thesis examiner: Prof. Dr. Didier Stricker
Thesis co-examiner: Prof. Dr. Paul Lukowicz

Date of Submission: 18 September 2013
Date of Defense: 9 January 2014

D386

Abstract

Regular physical activity is essential to maintain or even improve an individual's health. There exist various guidelines on how much individuals should do. Therefore, it is important to monitor performed physical activities during people's daily routine in order to tell how far they meet professional recommendations. This thesis follows the goal to develop a mobile, personalized physical activity monitoring system applicable for everyday life scenarios. From the mentioned recommendations, this thesis concentrates on monitoring aerobic physical activity. Two main objectives are defined in this context. On the one hand, the goal is to estimate the intensity of performed activities: To distinguish activities of light, moderate or vigorous effort. On the other hand, to give a more detailed description of an individual's daily routine, the goal is to recognize basic aerobic activities (such as *walk*, *run* or *cycle*) and basic postures (*lie*, *sit* and *stand*).

With recent progress in wearable sensing and computing the technological tools largely exist nowadays to create the envisioned physical activity monitoring system. Therefore, the focus of this thesis is on the development of new approaches for physical activity recognition and intensity estimation, which extend the applicability of such systems. In order to make physical activity monitoring feasible in everyday life scenarios, the thesis deals with questions such as 1) how to handle a wide range of *e.g.* everyday, household or sport activities and 2) how to handle various potential users. Moreover, this thesis deals with the realistic scenario where either the currently performed activity or the current user is unknown during the development and training phase of activity monitoring applications. To answer these questions, this thesis proposes and developes novel algorithms, models and evaluation techniques, and performs thorough experiments to prove their validity.

The contributions of this thesis are both of theoretical and of practical value. Addressing the challenge of creating robust activity monitoring systems for everyday life the concept of other activities is introduced, various models are proposed and validated. Another key challenge is that complex activity recognition tasks exceed the potential of existing classification algorithms. Therefore, this thesis introduces a confidence-based extension of the well known AdaBoost.M1 algorithm, called Conf-AdaBoost.M1. Thorough experiments show its significant performance improvement compared to commonly used boosting methods. A further major theoretical contribution is the introduction and validation of a new general concept for the personalization of physical activity recognition applications, and the development of a novel algorithm (called Dependent Experts) based on this concept. A major contribution of practical value is the introduction of a new evaluation technique (called leave-one-activity-out) to simulate when performing previously unknown activities in a physical activity monitoring system. Furthermore, the creation and benchmarking of publicly available physical activity monitoring datasets within this thesis are directly benefiting the research community. Finally, the thesis deals with issues related to the implementation of the proposed methods, in order to realize the envisioned mobile system and integrate it into a full healthcare application for aerobic activity monitoring and support in daily life.

Acknowledgments

Many have supported, influenced and helped me in the process which ultimately resulted in this thesis. First, I would like to thank Prof. Dr. Béla Pataki from the Budapest University of Technology and Economics, whose classes on topics of machine learning greatly inspired me. My special interest in ensemble learners originates from this time, which led to arguably the most important contributions of this thesis.

Over the course of my thesis I have submitted papers to various conferences, receiving a good amount of scientific reviews of my work. Many of these reviews were quite helpful by providing constructive criticism, which often led to new ideas. Therefore, I would like to thank all the anonymous reviewers of these conferences. Furthermore, I would like to thank the organizers and participants of the *Workshop on Robust Machine Learning Techniques for Human Activity Recognition* held at SMC 2011, which was a truly inspiring event for me.

For evaluation purposes I mainly used two datasets throughout my thesis, namely the PAMAP and PAMAP2 datasets. These datasets were recorded from co-workers and students at DFKI. I would like to thank all the anonymous volunteers participating in these data recordings – and I am sorry to make you guys iron my shirts under scientific pretences! Moreover, I would like to thank my students Benjamin Schenkenberger and Markus Gräb for their help in the development of the physical activity monitoring system prototypes. Furthermore, I would like to thank Vladimir Hasko for providing me with various illustrations.

I would like to thank my supervisor, Prof. Dr. Didier Stricker, the opportunity to carry out my research work. I would also like to thank my other two committee members, Prof. Dr. Paul Müller for accepting the role of chair of the committee, and Prof. Dr. Paul Lukowicz for agreeing to be the co-examiner of my thesis.

My very special thanks goes to Dr. Gustaf Hendeby, who supported me in countless ways over the course of the thesis. His way of being critical but always constructive and paying attention to the smallest details helped me in different aspects of performing rigorous scientific work. Gustaf, I thank you for our stimulating discussions, your countless advice, valuable feedback and always taking interest in my work! I also thank for all the practical help over the years, helping out with my hardware problems, being my personal LaTeX, git, C++, etc. expert and dealing with my annoying questions, or even providing medical service at midnight if needed. I am also grateful for you proof-reading my thesis and this way improving its quality. Overall, I believe that several really good papers show the result of our fruitful cooperation – and hope to continue this in the future!

During the time of being Ph.D. candidate I was researcher in the Augmented Vision group at DFKI, Kaiserslautern. I would like to thank many of my former colleagues there for all the activities which meant a welcoming distraction from the hard and stressful work of a scientist, such as bouldering, playing squash, soccer or billiards, or just enjoying a nice cup of hot chocolate from the fourth floor vending machine. In particular I would like to thank Leivy Michelly Kaul for helping out with all the administrative challenges and always having a friendly word for me. I would also like to thank Christiano Gava, my long-term weekend buddy, whose pres-

ence made all the Saturdays and/or Sundays spent at work less monotonous. My very special thanks goes to Sarvenaz Salehi, who made the last year I have spent with my thesis, including the entire writing process, so much more enjoyable. Azizam, I am also thankful for the great doctoral hat, a truly personal gift with all the memories from this time!

Last but not least I would like to thank my family. I would like to thank my little brother Tibor (sorry, Dr. Tibor Reiss) who received his doctoral degree way before me. This embarrassing fact was highly motivating me to finish my thesis as soon as possible. Now it's done – this means no more jokes about it anymore! I would also like to thank my parents. Without their support during and beyond the thesis none of this would have been possible. Therefore, I would like to dedicate my thesis to my parents.

Kaiserslautern, January 2014
Attila Reiss

Contents

1

Introduction

Regular physical activity is essential to maintain or even improve an individual's health. There exist various guidelines on how much individuals should do. Therefore, there is a need to monitor performed physical activities in order to compare them to professional recommendations. For a long time, questionnaires about the individual's physical activity practice represented the main choice of clinical personnel, resulting in a highly imprecise control. However, with recent progress in wearable technology, unobtrusive mobile long-term physical activity monitoring has become reasonable.

The overall goal of this thesis is the development of a physical activity monitoring system, with two main objectives. On the one hand, the goal is to monitor how far individuals meet professional recommendations. Concentrating on aerobic activity, this means the intensity estimation of performed activities: to distinguish activities of light, moderate and vigorous effort. On the other hand, to give a more detailed description of an individual's daily routine, the goal is to recognize basic aerobic activities and basic postures.

Since the technological tools to create the envisioned physical activity monitoring system largely exist nowadays, the focus of this thesis is on developing methods to extend the applicability of such systems. In order to make physical activity monitoring feasible in everyday life scenarios, the thesis deals with questions such as 1) how to handle a wide range of *e.g.* everyday, household or sport activities and 2) how to handle various potential users. Moreover, this thesis deals with the realistic scenario where either the currently performed activity or the current user is unknown during the development and training phase of activity monitoring applications. To answer these questions, this thesis proposes and developes novel algorithms, models and evaluation techniques, and performs thorough experiments to prove their validity.

This chapter first presents facts on overweight and obesity in Section 1.1, followed by defining the motivation of work performed within this thesis in Section 1.2. Section 1.3 defines the key challenges addressed in this thesis. Section 1.4 briefly describes the contributions, each presented in the following chapters of this work. Finally, Section 1.5 gives an outline of the thesis.

1.1 Background

According to the World Health Organization (WHO) the number of overweight and obese people increases rapidly [191]. Increased body mass index (BMI) is a major risk factor for medical conditions such as diabetes, cardiovascular diseases, musculoskeletal disorders and certain types of cancer. This makes overweight and obesity the fifth leading risk factor for global deaths.

Overweight and obesity is caused by an energy imbalance between calories consumed (through food and beverages) and calories expended (through *e.g.* physical activity). The two main factors according to WHO are [191]:

- An increased intake of energy-dense food that are high in fat, salt and sugars but low in vitamins, minerals and other micronutrients.
- A decrease in physical activity due to the increasingly sedentary nature of many forms of work, changing modes of transportation and increasing urbanization.

Recent studies suggest that the increasing number of overweight and obese people is more driven by a reduction in energy expenditure than by a rise in energy intake. The key facts given by the WHO fact sheet on obesity and overweight are the following (*cf.* [191], the fact sheet was last updated in March 2013):

- Worldwide obesity has more than doubled since 1980.
- 65% of the world's population live in countries where overweight and obesity kills more people than underweight (the fifth leading risk factor for global deaths).
- In addition, 44% of the diabetes burden, 23% of the ischaemic heart disease burden and between 7% and 41% of certain cancer burdens are attributable to overweight and obesity.
- More than 40 million children under the age of five were overweight in 2010.
- Overweight and obesity are preventable. Beside a balanced diet the engagement in regular physical activity is a key element in reducing an individual's BMI.

1.2 Motivation

In response to the above presented facts, regular physical activity is essential. Its importance has been proven, there exist various guidelines and recommendations on how much individuals should do. Therefore, this section will argue that there is a need for monitoring individual's physical activities in their daily routine. Moreover, this section will show that with recent progress in wearable sensing and computing the technological tools exist nowadays to create the envisioned physical activity monitoring systems.

1.2.1 The Need of Regular Physical Activity

The health benefits associated with regular physical activity have been investigated in many research studies over the last decades. Strong evidence has been found that physical activity indeed reduces the risk of many diseases, including diabetes, cardiovascular diseases and certain types of cancer. Alford [4] gives an overview of the most recent studies on this topic, and argues that – apart from not smoking – being physically active is the most powerful lifestyle choice individuals can make to improve their health. A list of all the health benefits, where strong or moderate evidence was found that they can be associated with regular physical activity is given *e.g.* in [92]. The major research findings are the following:

- Regular physical activity reduces the risk of many adverse health outcomes.
- Some physical activity is better than none.
- For most health outcomes, additional benefits occur as the amount of physical activity increases through higher intensity, greater frequency and/or longer duration.
- Most health benefits occur with at least 150 minutes of moderate-intensity physical activity per week, such as brisk walking. Additional benefits occur with more physical activity.
- Both aerobic (endurance) and muscle-strengthening (resistance) physical activity are beneficial.
- Health benefits occur for children and adolescents, young and middle-aged adults, older adults, and those in every studied racial and ethnic group.
- The health benefits of physical activity occur for people with disabilities.
- The benefits of physical activity far outweigh the possibility of adverse outcomes.

There exist various recommendations on how much physical activity individuals should perform. The main idea behind these guidelines is that regular physical activity over months and years can produce long-term health benefits. However, physical activity must be performed each week to achieve these benefits [92]. The original recommendation of minimum 30 minutes per day of moderate intensity physical activity has been recently updated and refined [66]. According to the updated recommendation statement, "to promote and maintain health, all healthy adults aged 18 – 65 yr need moderate-intensity aerobic physical activity for a minimum of 30 min on five days each week or vigorous-intensity aerobic activity for a minimum of 20 min on three days each week. Also, combinations of moderate- and vigorous-intensity activity can be performed to meet this recommendation." In addition to aerobic activity, Haskell et al. [66] also recommend muscle-strengthening activity for at least twice a week. Moreover, they also point out that greater amounts of activity provide additional health benefits. Similar recommendations specifically for elderly are given by

Nelson et al. [113]. Important differences compared to the recommendations given in [66] are that they take into account the elderly's aerobic fitness, they recommend activities that maintain or increase flexibility, and balance exercises are recommended for older adults at risk of falling. Furthermore, Leavitt [92] gives key guidelines for different age groups separately: for children/adolescents, for adults and for elderly.

Apart from the above mentioned general guidelines, physicians could also give specific recommendations (*e.g.* as part of a custom care plan) to individuals. However, in both cases it is important to monitor how much activity individuals perform when unsupervised (*e.g.* at home or when carrying out their daily routine), to be able to tell how far they meet the recommendations. This is the main motivation of this thesis, focusing only on the recommendations given for aerobic physical activity.

The different guidelines mostly instruct to perform a certain amount of moderate- or vigorous-intensity aerobic activities. Concrete examples of activities of these intensity levels are given *e.g.* in the recommendations of Haskell et al. [66]: A table of common physical activities classified as light, moderate or vigorous intensity is presented[1]. For simplicity and availability reasons, there are a few traditionally recommended aerobic activities: walking, cycling, running and – in certain countries such as Germany – Nordic walking. Together with the postures lying, sitting and standing, most of an individual's daily routine can be described from the physical activity point of view. Therefore, the recognition of these activities and postures is essential in a system for aerobic activity monitoring.

The overall goal of this thesis is the development of a physical activity monitoring system, with two main objectives. On the one hand, the system aims to support the monitoring of an individual's daily routine to be able to tell in what way the individual meets the recommendations of *e.g.* [66] on aerobic activity. For this purpose, the system should classify miscellaneous activities performed by the individual according to their intensity level – in respect of the aforementioned guidelines – as activities of light, moderate or vigorous effort (intensity estimation task). On the other hand, to give a more detailed description of an individual's daily routine, the system should identify with a high reliability the aerobic activities traditionally recommended and the basic postures (activity recognition task).

1.2.2 The Tools Provided by Wearable Technology

The previous subsection concluded that there is a need for monitoring individuals' physical activities in their daily routine. Until recently questionnaires were the main choice of clinical personnel to assess how much and what type of activities their patients performed. However, this method is a clearly imprecise control of an individual's physical activity practice. With recent progress in wearable technology, unobtrusive and mobile activity monitoring has become reasonable. Therefore, this subsection will argue that the technological tools nowadays exist to create the previously defined physical activity monitoring system.

[1] This table uses the Compendium of Physical Activities [1] as the source of the metabolic equivalent (MET) of different activities, which is a common reference in the field of energy expenditure estimation of physical activity.

The progress of wearable computing can be followed when examining the contributions of the yearly IEEE International Symposium on Wearable Computers (ISWC), the arguably most important conference in this research field. Since the first ISWC conference (held in October 1997) most of the key topics of this field advanced tremendously, as pointed out by Thomas [175].

On the one hand, wearable sensing advanced in many ways. Nowadays small, lightweight, low-cost and accurate sensor units are commercially available, supporting wireless data transfer, internal data storage, etc. With this progress it becomes feasible for individuals to wear various sensor units all day. Further miniaturizing the sensors, integrating them into worn devices (*e.g.* the concept of a smart watch, *cf.* the eZ430-Chronos system from Texas Instruments [44] or the Sony SmartWatch [163]) and garment integration (the concept of e-textiles, presented *e.g.* in [26]) will result in the completely unobtrusive wearing of sensors.

On the other hand, with the appearance of smartphones, the original goals set for wearable computers were even exceeded [175]. With the smart phone technology a pervasive control unit is widely available, providing also a large amount of computation and graphics power to individuals. Different sensors integrated in smartphones have reached the quality to *e.g.* monitor the movement of their owners. Moreover, current mobile operating systems (*e.g.* Android) ensure a comfortable way of developing applications for smartphone-based solutions.

Overall, with the presented advances in wearable sensing and wearable computing, the technological tools exist to develop a mobile, unobtrusive and accurate physical activity monitoring system. Therefore, the realization of long-term monitoring of individuals' physical activities while performing their daily routine – the goal set and motivated in the previous subsection – has become feasible.

1.3 Problem Statement

The development of a physical activity monitoring system, supporting the recognition of performed activities and the assessment of their intensity level, has been motivated in the previous section. Moreover, Section 1.2.2 argued that the creation of such systems is feasible in an unobtrusive way, based on current smartphones and miniaturized wearable sensors. Therefore, the focus of this thesis is on developing methods for physical activity recognition and intensity estimation, which are applicable to available hardware components.

The recognition of basic physical activities (such as walk, run or cycle) and basic postures (lie, sit, stand) is well researched [42, 93, 100, 117], and is possible with just one 3D-accelerometer. Moreover, the intensity estimation of these basic activities has been the focus of recent studies, *e.g.* in [118, 173]. However, since these approaches only consider a limited set of similar activities, they only apply to specific scenarios. Therefore, one of the key challenges in the research field of physical activity monitoring is to not only include these traditional basic activities and postures, but also other activities: examples of *e.g.* everyday, household or sport activities.

There are countless number of physical activities (*e.g.* 605 different activities are listed in [1]), thus it is not feasible to recognize all of them. Moreover, in practice, activity monitoring systems usually focus on only a few activities. Nevertheless, all the other activities should not be completely ignored, but different solutions should be investigated to deal with them, in order to enhance the applicability of developed systems. On the one hand, this includes the investigation of how to model these other activities in classification problems defined for activity monitoring tasks. On the other hand, proper evaluation techniques should be introduced to deal with this issue, to simulate the effect of (known or unknown) other activities.

Compared to when only dealing with basic activities and postures, the introduction of a large number of other activities clearly increases the complexity of the activity recognition and intensity estimation tasks. Therefore, it should be investigated how well existing classification approaches perform on these tasks. In case the desired accuracy can not be reached, novel algorithms should be developed. In order to evaluate the proposed methods, proper datasets – including a wide range of physical activities – would be required. However, in the field of physical activity monitoring there is a lack of such commonly used, standard datasets. Therefore, the thesis will also address this issue by creating and releasing such datasets, and by benchmarking various activity monitoring problems.

Another key challenge addressed in this thesis is related to the fact that activity monitoring systems are usually trained on a large number of subjects, and then used by a new subject from whom data is not available in the training phase. Moreover, there is a high variability – concerning *e.g.* age, weight or physical fitness – of potential users, thus individual accuracy can vary a lot. Therefore, personalization approaches for activity recognition have become a topic of interest recently. However, existing solutions have several practical limitations, concerning *e.g.* computational time or their applicability for complex classification tasks. The goal of this thesis is to overcome these limitations by developing a fast and accurate personalization approach for mobile physical activity recognition applications.

Altogether, the overall goal of this thesis can be refined: To develop a mobile, personalized physical activity monitoring system applicable for everyday life scenarios. The next chapters will present the proposed methodology to deal with the here discussed challenges, and how these proposed algorithms can be realized as part of a state-of-the-art mobile activity monitoring application. The system created this way can be used with less constraints under realistic, everyday conditions than systems presented in previous, related work.

1.4 Contributions

This section briefly describes the contributions presented in each of the following chapters of this thesis. Moreover, this section provides with the information where these contributions have been published.

Chapter 3 addresses the lack of a commonly used, standard dataset in the field of physical activity monitoring. Two new datasets (the PAMAP and the PAMAP2 dataset) are created and both made publicly available for the research community.

The PAMAP dataset is first described in the conference paper [133],

> Attila Reiss and Didier Stricker. Towards global aerobic activity monitoring. In *Proceedings of 4th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, Crete, Greece, May 2011.

The PAMAP2 dataset is introduced in [136],

> Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *Proceedings of IEEE 16th International Symposium on Wearable Computers (ISWC)*, pages 108–109, Newcastle, UK, June 2012.

and described in more detail in [135],

> Attila Reiss and Didier Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of 5th Workshop on Affect and Behaviour Related Assistance (ABRA)*, Crete, Greece, June 2012.

The subsequent chapters heavily rely on both new datasets, using them for the evaluation of proposed methods. Moreover, there has been a certain impact in the research community by making these datasets publicly available, *cf.* Section 3.5.

Chapter 4 addresses the lack of established benchmarking problems in the field of physical activity monitoring. A benchmark is given, using a complete data processing chain (DPC) and comparing commonly used classification algorithms on a set of defined physical activity monitoring tasks. The benchmark shows the difficulty of different classification problems and reveals some of the challenges in this research field. The description of the benchmark and results are given in [135, 136]. The applied DPC is first described in [133], an extended description is given in the journal paper [137],

> Attila Reiss and Didier Stricker. Aerobic activity monitoring: towards a long-term approach. *International Journal of Universal Access in the Information Society (UAIS)*, March 2013.

Chapter 5 addresses a usually neglected point of view in the development of physical activity monitoring systems: It creates the means for simulating everyday life scenarios. This chapter focuses on the one hand on the subject independency of activity monitoring systems. The effect of applying subject dependent and subject independent evaluation techniques is first investigated in [139],

> Attila Reiss, Markus Weber, and Didier Stricker. Exploring and extending the boundaries of physical activity recognition. In *Proceedings of 2011 IEEE International Conference on Systems, Man and Cybernetics (SMC), Workshop on Robust Machine Learning Techniques for Human Activity Recognition*, pages 46–50, Anchorage, AK, USA, October 2011.

Further results on this matter are shown in the benchmark of [135, 136], providing evidence that overall subject independent validation techniques should be preferred for physical activity monitoring.

On the other hand, Chapter 5 focuses on including various other activities in activity monitoring classification tasks. Different models are proposed and evaluated, as described in the conference paper [142],

> Attila Reiss, Gustaf Hendeby, and Didier Stricker. Towards robust activity recognition for everyday life: methods and evaluation. In *Proceedings of 7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, Venice, Italy, May 2013.

Moreover, a new evaluation technique is introduced in this chapter (leave-one-activity-out), in order to simulate when an activity recognition system is used while performing a previously unknown activity. Applying the proposed methods has two important benefits compared to previous related work. First of all it is estimated how a system behaves in various everyday life scenarios, while this behaviour would be otherwise undefined. Second, the best performing methods and algorithms can be selected during the development phase of the system, with the desired generalization characteristic. Therefore, it is possible to develop a robust physical activity recognition system, as justified by the detailed results given in [142].

Chapter 6 addresses one of the main challenges revealed by the benchmark of [135, 136]: The difficulty of the more complex activity monitoring classification problems (caused by *e.g.* the introduction of other activities into these tasks) exceeds the potential of existing classifiers. Therefore, this chapter introduces a confidence-based extension of the well-known AdaBoost.M1 algorithm, called ConfAdaBoost.M1. The new algorithm is evaluated on various benchmark datasets (inside and outside of the research field of physical activity monitoring), comparing it to the most commonly used boosting techniques. Results show that ConfAdaBoost.M1 performs significantly best among these algorithms, especially on the larger and more complex physical activity monitoring problems. The ConfAdaBoost.M1 algorithm, along with the mentioned thorough evaluation, is presented in the conference paper [143],

> Attila Reiss, Gustaf Hendeby, and Didier Stricker. Confidence-based multiclass AdaBoost for physical activity monitoring. In *Proceedings of IEEE 17th International Symposium on Wearable Computers (ISWC)*, Zurich, Switzerland, September 2013.

Moreover, ConfAdaBoost.M1 is applied on a further dataset, created for human activity recognition on smartphones. The new classification method outperforms commonly used classifiers on this dataset as well, as presented in the paper [141],

> Attila Reiss, Gustaf Hendeby, and Didier Stricker. A competitive approach for human activity recognition on smartphones. In *Proceedings of 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Bruges, Belgium, April 2013.

Chapter 7 addresses another challenge revealed by the benchmark of [135, 136]: The diversity in how individuals perform different physical activities, and the so caused large variance in per-user accuracy of activity recognition applications. This chapter introduces a novel general concept for the personalization of such applications. An important benefit of the proposed concept is its low computational cost compared to other approaches, making it also feasible for mobile activity recognition applications. Moreover, a novel algorithm (called Dependent Experts) is introduced in this chapter, further increasing the performance of the personalized system. Both the proposed general concept and the new algorithm are evaluated on activity recognition classification tasks defined on the PAMAP2 dataset. A description of the novel general concept for personalization and the algorithm Dependent Experts, and results of the thorough evaluation are all given in the conference paper [138],

> Attila Reiss and Didier Stricker. Personalized mobile physical activity recognition. In *Proceedings of IEEE 17th International Symposium on Wearable Computers (ISWC)*, Zurich, Switzerland, September 2013.

Chapter 8 presents several contributions towards creating a mobile, unobtrusive physical activity monitoring system. First of all the idea of a modular activity monitoring system is presented, where different sets of sensors are required for different activity recognition and intensity estimation tasks. This idea was first described and justified with practical experiments in the publication [134],

> Attila Reiss and Didier Stricker. Introducing a modular activity monitoring system. In *Proceedings of 33rd Annual International IEEE EMBS Conference*, pages 5621–5624, Boston, MA, USA, August-September 2011.

A further contribution of Chapter 8 is a set of empirical studies showing that more complex, meta-level classifiers (boosted decision trees as a concrete example) are feasible and thus a considerable choice for mobile applications, there are no limitations regarding the computational costs. Part of these experiments is presented in [139].

Finally, this chapter describes the integration of the mobile system into a full healthcare application for aerobic activity monitoring and support in daily life. A key benefit of such an integrated system is the possibility to give feedback to both the patient (preserving or even increasing the motivation to follow a defined care plan) and the clinical personnel (providing valuable information on program adherence). The integrated mobile system is described in the conference paper [140],

> Attila Reiss, Ilias Lamprinos, and Didier Stricker. An integrated mobile system for long-term aerobic activity monitoring and support in daily life. In *Proceedings of 2012 International Symposium on Advances in Ubiquitous Computing and Networking (AUCN)*, Liverpool, UK, June 2012.

1.5 Thesis Outline

This thesis is organized in the following way:

Chapter 1 Introduction (this chapter) Motivates research work in the field of physical activity monitoring, defines challenges related to this topic and lists the contributions of this thesis.

Chapter 2 Related Work Gives an overview of human activity monitoring research, addressing four major topics: the type of monitored activities, applied sensing modalities, different machine learning methods and various application areas.

Chapter 3 Datasets for Physical Activity Monitoring Introduces two new datasets, recorded from a reasonable number of subjects performing a wide range of physical activities. Both datasets include ground truth and are publicly available to the research community.

Chapter 4 Data Processing and Classification Presents a data processing chain, describing the steps feature extraction and classification in more detail. It also introduces a benchmark, comparing commonly used classification algorithms on different physical activity monitoring tasks.

Chapter 5 Robust Activity Monitoring for Everyday Life: Methods and Evaluation Investigates the development and evaluation of robust methods for everyday life scenarios, with focus on the tasks of aerobic activity recognition and intensity estimation.

Chapter 6 Confidence-based Multiclass AdaBoost Proposes a confidence-based extension of the well-known AdaBoost.M1 algorithm, called ConfAdaBoost.M1. It also presents a large number of experiments, confirming that the novel algorithm outperforms existing boosting methods.

Chapter 7 Personalization of Physical Activity Recognition Introduces a novel general concept for the personalization of physical activity recognition applications. It also presents a novel algorithm based on this concept, and shows its benefits over existing approaches.

Chapter 8 Physical Activity Monitoring Systems Presents a modular, mobile activity monitoring system which implements methods introduced in the previous chapters. It also describes the integration of the mobile system into a full healthcare application for aerobic activity monitoring and support in daily life.

Chapter 9 Conclusion Summarizes the thesis, draws conclusions and gives ideas for possible future extension of the presented research work.

Appendix A Abbrevations and Acronyms

Appendix B Datasets: Supplementary Material Presents supplementary material related to the two introduced physical activity monitoring datasets.

2

Related Work

2.1 Introduction

This thesis focuses on monitoring the user's physical activity, one of the main topics of context awareness. The term context-aware computing was first introduced by Schilit et al. [156], with the goal to promote and mediate users' interaction with computing devices and other people, and to help navigate in unfamiliar places. For this purpose, situational and environmental information is used. Therefore, the user's activity and location are essential to provide the user with relevant information about his current context (activity recognition and location awareness).

With recent advances in hardware technology related to mobile sensing and computing, pervasive and ubiquitous computing has evolved tremendously. In the past decade, a vast amount of research has been performed in these research areas, resulting in various solutions and applications affecting our everyday life. Therefore, this chapter gives a general overview of recent, state-of-the-art research related to activity monitoring. Section 2.2 categorizes the wide range of activities which have been monitored and recognized in related work. Section 2.3 describes the different sensing modalities that have been used to monitor these activities. Section 2.4 reviews machine learning methods applied for activity recognition in related work. Finally, Section 2.5 describes and categorizes applications based on the results of research performed for activity monitoring and recognition.

The rest of this thesis will focus on a very specific case, on a fraction of the here presented activity monitoring research: The monitoring of low-level physical activities with wearable sensors, with the intention to be used mainly for fitness and healthcare applications. Therefore, Section 2.6 defines where this thesis is positioned with respect of the major topics discussed in this chapter. Moreover, it should be noted that related work directly relevant to the different contributions of this thesis will be presented in the respective chapters.

2.2 Activities

There exist a wide range of activities that have been monitored and recognized in related work. Nevertheless, a definite and commonly used categorization of them is not provided in the literature. Huynh [74] presents a possible way to categorize activities by grouping them based on duration and complexity. This categorization defines 3 groups: gestures (brief and distinct body movements), low-level activities (sequence of movements or a distinct posture) and high-level activities (collection of activities). Gesture recognition is a large research topic in itself, but is not within the focus of this thesis. A brief overview of related work performed for low-level and high-level activity monitoring will be given in following subsections.

There are various classes of activities which can not be clearly grouped into the above defined 3 categories. A class of activities relevant for this thesis are the activities of daily living (ADL). A brief overview of existing literature related to these activities is presented in Section 2.2.3. Further important topics of activity monitoring research are *e.g.* fall detection [24, 67, 95, 104, 151], sleep monitoring (wake-sleep patterns, quality of sleeping) [23, 150, 183], the recognition of workshop or assembly activities [101, 165, 188], etc. Since these latter topics are outside of the scope of this thesis, they are not further investigated here.

2.2.1 Low-Level Activities

The monitoring and recognition of low-level activities is well researched. It has been shown that reliable recognition of a few activities (usually locomotion activities and postures) is possible with just one sensor, a 3D-accelerometer. An example is given by Lee et al. [94], where only a tri-axial accelerometer is used to create a real-time personal life log system, based on activity classification. The authors selected 7 activities to be distinguished: *lying*, *sitting*, *standing*, *walking*, *going upstairs*, *going downstairs* and *driving*. Further examples of recognizing a few low-level activities with just one 3D-accelerometer are given in [42, 93, 100, 117].

A nowadays popular research topic is the monitoring of low-level activities while only using sensors provided by smartphones. For example, Kwapisz et al. [86] recognize the activities *walking*, *jogging*, *going upstairs*, *going downstairs*, *sitting* and *standing* by using the embedded accelerometer of Android-based cell phones. However, using mobile phones for activity monitoring generates several new challenges. For example, both the position (in user's hand, in user's pocket, in a bag carried by the user, etc.) and the orientation of the device is not determined during regular usage, which has to be taken into account during the training phase of such applications [16]. Another key challenge is the fact that activity monitoring applications constantly drain the battery of mobile phones, which could limit the regular use of these devices. Therefore, energy-efficient solutions have recently been investigated for activity recognition on mobile phones, presented *e.g.* in [62, 196].

A further topic of interest nowadays is the recognition of a wide range of low-level activities with multiple sensors, placed on multiple locations of the user's body. For example, Patel et al. [122] use 10 Shimmer nodes (each including an accelerometer and a gyroscope) [160] to distinguish different activities performed during gym exer-

cising and the user's daily routine. Another example is presented in [8]: 19 activities (mainly aerobic sport activities, such as *cycling*, *rowing* or *exercising on a cross trainer*) are recognized using 5 inertial measurement units. Since this thesis mainly focuses on the monitoring of low-level activities (*cf.* Section 2.6), mostly on the recognition of a wide range of activities with multiple sensors, further examples of related work on this topic will be presented in the remaining chapters.

2.2.2 High-Level Activities

High-level activities are composed of a set of low-level activities, *e.g.* the high-level activity *shopping* can consist of *driving car*, *walking*, *standing*, etc. Moreover, high-level activities usually last longer than low-level activities, they can last up to a few hours. Although the recognition of high-level activities is important for the description of an individual's daily routine, research has so far mainly focused on low-level activities. Furthermore, with the knowledge obtained about an individual's high-level activities, the recognition of his low-level activities can be improved. For example, if the recognized high-level activity is *shopping*, during this time the probability of the activity *walking* is much higher than *Nordic walking*.

An example on how the recognition of low-level activities can be utilized for high-level activity recognition is shown by Huynh et al. [78]. They recorded a realistic dataset including 3 high-level activities: *preparing for work*, *going shopping* and *doing housework*. Each of these activities is composed of a set of low-level activities, for example *going shopping* consists of *driving car*, *walking*, *working at computer*, *waiting in line in a shop* and *strolling through a shop*. Both low-level and high-level activity labels are given in their recorded dataset. The authors used simple features and common algorithms (kNN, HMM, SVM). One of their key findings was that the recognition of high-level activities can be achieved with the same algorithms as the recognition of low-level activities. Moreover, they could distinguish between the 3 defined high-level activities with a recognition rate of up to 92%.

The work presented in [79] uses topic models to recognize daily routines as a probabilistic combination of activity patterns. First of all, the authors recognize a large set of low-level activities using a wearable sensor platform. By using this information, they show that the modeling and recognition of daily routines is possible without user annotation. Overall, 4 daily routines (high-level activities) were recognized: *commuting*, *office work*, *lunch routine* and *dinner activities*.

Finally, the Opportunity framework and dataset [103, 144] also provides the means to analyze how high-level activities can be composed of a set of recognized low-level activities. The dataset provides a sensor rich, kitchen like scenario. The work in [69] presents an approach based on the dynamic configuration of HMMs, evaluated using the Opportunity framework. The authors define and recognize 3 composite activities: *coffee making*, *coffee drinking* and *table cleaning*.

2.2.3 Activities of Daily Living

A specific set of activities, called activities of daily living, was first proposed by Katz et al. [84] in order to provide a standardized way to estimate the physical well-being

of elderly and their need for assisted living. The following activities are included in the set of ADLs: *bathing*, *dressing*, *toileting*, *transferring*, *continence* and *feeding*. Moreover, Lawton and Brody [91] proposed another set of activities, called instrumental activities of daily living (IADL), in order to assess how well elderly interact with the physical and social environment. The set of IADLs consists of the following activities: *using telephone*, *shopping*, *food preparation*, *housekeeping*, *doing laundry*, *transportation*, *taking medications* and *handling finances*.

Various approaches exist for the monitoring and recognition of specific subsets of ADLs/IADLs. For example, Stikic et al. [166] combine RFID tags and accelerometers to recognize 10 housekeeping activities (such as *dusting*, *ironing* or *vacuum cleaning*). With these sensing modalities they combine two main assumptions related to ADLs: 1) the objects people use during the execution of an activity robustly categorize that activity (RFID tags) and 2) the activity is defined by body movements during its execution (accelerometers). The authors of [106] use a wrist-worn device to distinguish 15 ADLs. This device includes the following sensors: accelerometer, microphone, camera, illuminometer and digital compass. Results show that the camera is the most important single sensor in the recognition of this subset of ADLs, followed by the accelerometer and the microphone. The work by Maekawa et al. [107] introduces the concept of mimic sensors: the mimic sensor node has the shape of objects like a AA battery or an SD memory card, and provides the functions of the original object. Moreover, these sensor nodes provide additional information (*e.g.* current flow of the device), which can be used to detect electrical events. These events can then be used to recognize ADLs such as *shaving* or *vacuum cleaning*. Further recent examples of research work on the topic of monitoring and recognizing ADLs/IADLs are presented *e.g.* in [30, 182, 197].

2.3 Sensors

A wide range of sensors have been investigated and used in related work of activity monitoring, from binary switches to cameras. Generally, two types of sensors can be distinguished in this field: wearable sensors (placed on the user) and ambient sensors (placed around the user). It mainly depends on the application what type and how many sensors should be used. However, the selection of sensors is important when developing an activity monitoring system, several factors should be considered:

- How intrusive the user experiences the sensor. In case of wearable sensors this means *e.g.* how comfortable they are, and in case of ambient sensors if they are in sight *e.g.* in a home environment.
- Privacy issues: how much and how sensitive information is recorded and stored by the sensor (one of the main concerns with sensors such as cameras or microphones).
- Ease of setup, *e.g.* in case of wearable sensors the user should be able to put on the sensors without external help and in a short time.

- Maintenance, *e.g.* battery time or how easy it is to repair/replace components deployed in a home environment.
- Cost factor: how expensive the entire setup is.
- What the sensor measures, thus the most activities should be differentiated with the provided sensor data.

This section presents the most commonly used sensing modalities applied in the literature of activity monitoring: inertial sensors, physiological sensors, cameras, audio sensors, sensors deployed in objects and radio-based sensing. Finally, the combination of complementary sensors applied in related work is discussed in the last subsection.

2.3.1 Inertial Sensors

Inertial measurement units, and especially accelerometers (*cf.* Section 4.1) are probably the most broadly used sensors for activity monitoring. These sensors are inexpensive, small, lightweight, can be characterized with relatively low energy consumption and are usually experienced as less intrusive than many other sensors. Moreover, high recognition rates can be achieved with inertial sensors on *e.g.* ambulation or sport activities. For example, Altun et al. [8] recognize 19 physical activities by using 5 inertial measurement units, placed on the subject's torso and left/right arms/legs. Another example is presented in [38], where the authors use 5 tri-axial accelerometers to recognize a set of locomotion and everyday activities.

Ugulino et al. [179] give an overview of recent work in human activity recognition based on accelerometer data. Moreover, since this thesis mainly uses acceleration data for the monitoring of physical activities, a large amount of further examples will be presented in the following chapters concerning related work on this topic.

2.3.2 Physiological Sensors

Physiological sensors have been applied for the monitoring of human activities in various previous work. On the one hand, they are usually more expensive and more intrusive than inertial sensors. Moreover, contrary to *e.g.* acceleration data, physiological signals react slower to activity changes. For example, after performing physically demanding activities (*e.g. running*) the subject's heart rate remains elevated for a while, even if he is inactive (*e.g. lying* or *sitting*). On the other hand, physiological sensors are somewhat complementary to inertial sensors. For example, distinguishing between *walking* and *walking with load* is practically impossible with just inertial sensors, but a significant difference can be observed in selected physiological signals. Therefore, various related work proposes the combination of inertial and physiological sensing for activity monitoring, *cf.* Section 2.3.7. Moreover, this thesis adapts this concept by using acceleration and heart rate data together throughout the entire work.

In a large study carried out with wearable sensors, Pärkkä et al. [117] also investigated the use of various physiological sensors for human activity classification. The

following vital signs were included in their study: ECG (electrocardiogram), heart rate, respiratory effort, oxygen saturation, skin resistance and skin temperature. However, they found that physiological signals did not provide very useful data for activity recognition. They concluded that physiological signals correlate with the intensity level of performed activities, but they do not reflect the type of the activity. Moreover, Pärkkä et al. [117] observed larger interindividual difference in measured vital signs than *e.g.* in inertial signals, thus further limiting the applicability of physiological sensors.

The work presented by Lara et al. [89] comes to a different conclusion than [117]: The authors state that vital signs are indeed useful to discriminate between certain activities. They are using the BioHarness BT chest sensor strap [199], which measures several physiological attributes beside of 3D-acceleration: heart rate, respiration rate, breath amplitude, skin temperature and ECG amplitude. The authors apply structure detectors on the physiological signals, and propose two new features for vital signs: magnitude of change and trend. With these features, the discrimination between activities during periods of vital sign stabilization can be improved.

This paragraph mentions further examples of using different physiological sensors for activity monitoring. Respiration rate is measured and applied for the assessment of physical activities *e.g.* in [98, 112]. Features extracted from GSR signal (galvanic skin response) are good indicators to identify the presence of mental stress, even when the user performs different activities [170]. Haapalainen et al. [64] address a similar problem, the real-time assessment of cognitive load while the user is active, relying mainly on features extracted from GSR and ECG data. Finally, Chang et al. [29] use a non-contact portable heart rate monitor to predict driver drowsiness.

2.3.3 Image-based Sensing

The recognition of activities and gestures using external cameras has been the focus of extensive research. A survey on the recognition of human actions and activities from image data is given in [125]. However, using cameras for activity monitoring has several major issues. First of all, although video is very informative, automatically recognizing activities from video data is a complex task. A solution to this issue is to extract certain features from images, related to *e.g.* the location of the user or which objects in the environment are used. Such approaches could enable real-time recognition of various ADLs. An example is presented by Duong et al. [40]. Multiple cameras are installed in a room, observing a person performing different activities. The room is divided into squared regions, some of these including objects of interest (*e.g.* a stove). The multi-camera system tracks the person, returning a list of visited regions. This list can then be used to recognize actions such as *using the stove*.

A second major issue of camera-based activity monitoring systems is their lack of pervasiveness. The cameras are usually installed indoors, individuals have to stay within the field of view of the imaging sensors, defining a very strong limitation of the applicability of such systems. A way to overcome this limitation is to use wearable cameras instead of static ones, which is feasible due to recent miniaturization concerning hardware components. An example of this wearable vision concept is

presented in [172], where a camera is worn on the shoulder of the user, observing the interaction with various objects. Moreover, the wrist-worn device presented by Maekawa et al. [106] to recognize different ADLs, also includes a small camera complementing the other sensors. From this camera, a colour histogram is computed in each image. This information is used to determine how well a colour performs to distinguish a certain ADL class from other ADL classes.

A further major issue of using video sequences for human activity monitoring is privacy: Most individuals would have severe concerns about being permanently monitored and recorded by cameras. Wearable imaging sensors could at least partially solve this issue by simply enabling the user to turn off the sensor when monitoring and recording is undesired. Another way to ensure the privacy of the user is when a certain abstraction of raw image data is directly performed on the sensor device, and only this abstracted information is stored or used for further processing.

Despite the above described major issues and the fact that using cameras is an expensive way of monitoring activities, they can be used as a source of additional information to improve the performance of *e.g.* a wearable system. Bahle et al. [14] investigates this concept by using vision-based devices in the user's environment in an opportunistic way to improve wearable activity recognition. In case video data is available (*e.g.* the user is passing through a space observed by a camera), body motion information derived from the video signal is correlated with on-body sensor information. The goal is to improve the on-body system by *e.g.* determining the location of the sensors on the user's body.

2.3.4 Audio-based Sensing

Many human actions and activities produce characteristic sound. Therefore, activity monitoring based on audio sensing is a valid approach. There exist several examples in related work showing that the recognition of a well defined set of human activities is possible from just audio data. For example, Stork et al. [167] present a single microphone-based system recognizing 22 different sounds, corresponding to human actions and activities in a bathroom and kitchen context (*e.g. brushing teeth*, *boiling water* or *eating cornflakes*). They use mel-frequency cepstral coefficient (MFCC) features and a segmentation-free approach, and reach a recognition rate of over 85%.

The work in [197] presents a wearable acoustic sensor, called BodyScope, to record the sounds produced in the user's throat area and classify them into human activities. The sensor is attached to the user's neck and consists of a modified Bluetooth headset with a uni-directional microphone embedded into one of its earpieces. Since positioned on the side of the neck, the device amplifies the sounds produced inside the throat and minimizes audio from external sources. The BodyScope system is used to distinguish 12 different activities, such as *eating*, *drinking*, *speaking*, *laughing*, and *coughing*.

Another example of using a wearable acoustic sensor is presented by Zhan and Kuroda [200]. They rely on environmental background sound, which is a rich information source for identifying individual and social behaviours. The background sounds of 18 personal activities (*vacuum cleaning*, *shaving*, *drinking*, etc.) and 4 social

activities (*e.g. shopping* or *outside dining*) are distinguished. During these activities, the sensor node including the microphone was hung in front of the user's chest. The authors propose the use of Haar-like sound features and an HMM classifier, and claim an average recognition accuracy of nearly 97%.

Apart from using audio sensing for human activity monitoring, Rossi et al. [146] propose to apply audio data for context recognition. They discriminate a wide range of daily life situations, defined by objects (*e.g.* coffee machine, shaver), locations (*e.g.* office, restaurant) or animals and persons (*e.g.* dog, speech), all producing characteristic sound. In order to model these 23 sound context categories, they use crowdsourcing, thus the large amount of openly available audio samples annotated by various web users.

Although the above described approaches achieve promising results, human activity monitoring based on audio sensing has several limitations. First of all, the above mentioned results were mostly achieved under laboratory conditions. Under realistic settings, due to background noise, the performance of such systems is significantly lower, as pointed out *e.g.* in [197]. Moreover, while certain activities produce characteristic sound (*e.g.* typically ADLs), many other activities (*e.g.* different ambulation or sport activities) have no specific audio pattern. Nevertheless, using audio data in combination with other types of sensors is beneficial, as discussed in Section 2.3.7. Finally, similar to image-based sensing, privacy is a major issue when using audio sensors. A solution could be to compute sound-features directly on the wearable device, as proposed by [106].

2.3.5 Object Use

Many activities, especially ADLs/IADLs, can be characterized by the objects the user interacts with while performing that activity. Therefore, human activity monitoring based on object use is a well researched topic. The most typical approach is to instrument objects in the environment with RFID tags, and use data from a wearable RFID tag reader. An example of realizing this concept is presented by Philipose et al. [123]. They tag objects of interest in a home environment, and equip the user with a glove which includes an embedded RFID reader. From this setting it is possible to detect when and which objects the user interacts with. This provides useful information to recognize user activities such as food preparation or personal hygiene. In total 14 ADLs were chosen to be monitored by this system. For each of these activities a probabilistic model is created, which uses observations of object usage as input.

Another approach based on inferring activities from object use is presented in [180]. This work relies on simple sensors with discrete or even binary output, such as contact switches used to monitor the open/closed state of doors or cupboards. A sensor network consisting of wireless sensor nodes is created in a home setting. The sensor readings are utilized by temporal probabilistic models to recognize several ADLs, such as *showering* or *having breakfast*.

Either instrumenting objects with RFID tags or deploying a wireless sensor network of binary switches, the installation and maintenance of such systems usually entails high costs. Moreover, the attached RFID tags or sensor nodes might lower the

aesthetics of objects in the home. A solution to overcome these issues is to apply the mimic sensors proposed in [106] (*cf.* Section 2.2.3). Another issue of activity monitoring based on object use is that this concept is restricted to mainly home settings, but is not feasible otherwise. Moreover, activities not requiring object interaction can not be dealt with relying only on this concept. Therefore, to recognize all kind of activities of individuals' daily routine (including also *e.g.* locomotion or sport activities), a combination with other types of sensors is required, as suggested in Section 2.3.7.

2.3.6 Radio-based Sensing

An interesting, relatively new field of activity monitoring is device-free radio-based activity recognition (DFAR). The definition of a DFAR system is the following: A system which recognizes the activity of a person using analysis of radio signals while the person itself is not required to carry a wireless device [158]. The basic idea of DFAR is that if a human moves between transmitter and receiver wireless nodes, nearby receivers will show fluctuations in the received signals' power. The parameter describing this signals' power is the RSSI (received signal strength indicator) value. Therefore, the feature RSSI variance can be used to detect human movement.

Only few works exist related to DFAR. An example of such a system is presented by Scholz et al. [157]. They use two software defined radio nodes, placed to the right and left side of an office door. One node is configured to send a continuous sine signal on the 900 MHz band, and the other node receives and analyzes this signal. With this setup, the activities *walking* and *talking on the mobile phone* can be recognized, and the state of the office door (open/closed) can be determined.

While initial results achieved with DFAR systems are promising, this concept has several major drawbacks. For example, the application of the system is restricted to indoor environments. Moreover, current approaches can only recognize very few activities, and are limited to a single user. Nevertheless, the convenient setup for the user (no wearable sensors required, basically no privacy issues) motivates further investigation of this idea. An introduction into DFAR and the description of the current status of this field is given in [158].

2.3.7 Combination of Different Types of Sensors

As discussed in the previous subsections, each of the here presented sensing modalities have their benefits and drawbacks. Moreover, these sensors are complementary to each other to some extent. Therefore, a combination of two or more types of sensors might be beneficial for activity monitoring applications. This subsection will give examples of different combinations of sensors applied in related work.

Adding information about user's location could improve existing activity monitoring systems. For example, when outdoors, GPS information could be used to derive the user's speed, which would help to distinguish *e.g.* between *walking* and *running*. An approach to estimate user's low-level activities and spatial context is presented in [169], using GPS and a set of wearable sensors. Indoors, information about *e.g.* in which room the user is located when performing a certain activity could be of interest. An example application is presented by Chen et al. [30]. Information about the user's

location (sub-areas, such as kitchen or living room, are defined in a smart home) is used to improve the recognition rate of a system which uses RFID tags to recognize ADLs performed by elderly. They investigate two different ways to include the location information into their existing system: by introducing a new feature based on the location information, and by using location information to filter out irrelevant sensor readings.

The combination of inertial data and RFID tags is presented by Stikic et al. [166] to recognize 10 housekeeping activities. The combination of inertial sensors and microphones is used by Lukowicz et al. [101] to recognize workshop activities. Finally, the combination of inertial and physiological sensors has been used successfully for physical activity monitoring. For example, Crouter et al. [36] show that combining acceleration and heart rate data improves on the intensity estimation of performed activities compared to when only using inertial data. Moreover, the combination of accelerometers and a heart rate monitor will be used throughout this thesis for physical activity recognition and intensity estimation.

As a final note of this section, it should be noted that integrating different types of sensors into one device is clearly beneficial over deploying them separately. This statement is especially true for long-term wearable sensing, where the user's comfort should be taken into account. For this reason, modern smartphones are clearly interesting for activity monitoring applications, especially if sensors are required where the orientation and location of the device in respect to the user's body is not crucial. Most state-of-the-art smartphones provide with a long list of sensors: camera, GPS, accelerometer, gyroscope, microphone, compass, ambient light sensor, proximity sensor, etc. A survey on mobile phone sensing can be found *e.g.* in [88].

2.4 Learning Methods

This section gives an overview of the wide range of machine learning methods applied for the recognition of human activities. Typically, data samples recorded in an activity monitoring system are transformed into feature vectors, which are then used as input for training a classifier (*cf.* Section 4.2). The appropriate learning approach depends on many factors, *e.g.* the type of activities to be classified or the type of recorded data. Dependent on whether labeled training data is available or not, the distinction between supervised and unsupervised learning methods can be made.

Supervised learning approaches have been the most common choice in the literature of activity monitoring. These methods require annotations (ground truth) along with the recorded raw sensory data for training. Commonly used supervised classification methods in the field of human activity monitoring are the following:

- Decision tree classifiers, including custom decision trees [15, 43, 117] and various automatically generated decision tree algorithms (C4.5, ID3) [15, 61, 117].
- Bayesian classifiers, *e.g.* the Naive Bayes classifier [15, 61, 100].
- Instance based classifiers, such as the k-nearest neighbors (kNN) method [61, 108, 197].

- Artificial neural networks (ANN) [43, 63, 117].
- Support vector machines (SVM) [39, 98, 197].
- Markov models, including hidden Markov models (HMM) [68, 78, 182, 200] and conditional random fields (CRF) [171, 180].
- String-matching-based methods [165].
- Fuzzy logic-based classifiers [16, 93].

A comparison of different supervised classification approaches applied for activity recognition can be found *e.g.* in [8, 122]. Since supervised machine learning algorithms are one of the main focuses of this thesis, more information about these methods can be found in the subsequent chapters (*cf. e.g.* the data processing chain in Section 4.2). Moreover, some of the above listed classifiers have been used as part of an ensemble or meta-level classifier. Examples of ensemble learning algorithms used for activity recognition are boosting, bagging or plurality voting [131]. More information about these methods is given in Section 4.2.4, and specifically about different boosting variants in Chapter 6.

Unsupervised learning approaches are by far less commonly used for activity monitoring. These methods construct models directly from unlabeled data, using *e.g.* density estimation or clustering. Examples of unsupervised learning of different activities are presented *e.g.* in [31, 77, 96]. Finally, semi-supervised learning methods have been applied for human activity monitoring recently, delivering promising results. These methods combine a usually small amount of labeled data with large amounts of unlabeled data. Examples of realizing semi-supervised approaches for activity recognition are given in [3, 5, 37, 76].

2.5 Applications

This section discusses application areas of human activity monitoring. Related work presents different forms of activity recognition, and shows that these are broadly applicable. Lockhart et al. [99] give a survey on mobile activity recognition applications. They argue that little practical work has been done in the area of applications in mobile devices so far. Moreover, they define three major types of applications: those that benefit end users, those that benefit developers and third parties, and those that benefit crowds and groups. However, these types of applications are not mutually exclusive. Therefore, the following subsections will give a list and short description directly of different application areas.

2.5.1 Fitness, Sport

One of the most advanced application areas is the monitoring of fitness and sport activities. This application area can even show a wide range of commercially available products. The first such devices were traditional pedometers, which offer the assessment of features like distance traveled or calories burned. Most recent products

and applications utilize a broader range of sensors and provide with more detailed information, *e.g.* about the intensity and duration of performed physical activities, stairs climbed, etc. An example commercial product is the Fitbit system [50], which is a small chip-on device containing a 3D motion sensor, and provides the above described functionality. Concerning research performed in this area, there exist a large amount of related work on assessing the intensity (*e.g.* in [118, 187]) or recognizing the type of performed physical activities (*e.g.* in [28, 43]), or both [137, 173]. Moreover, it was also shown that by detecting the type of performed activities, the estimation of energy expenditure can be improved [2, 22].

Apart from applications monitoring an individual's physical activities in general, there exist work on monitoring specific sport activities. For example, Strohrmann et al. [168] investigate the potential of wearable sensors to derive kinematic features in running. With two miniature inertial measurement units, attached to the athlete's foot and hip, the authors could distinguish between experienced and unexperienced runners. Another example of monitoring a specific sport is given by Bächlin et al. [13], who analyze ski-jumping from on-body acceleration data. With sensors attached to the athlete's legs, arms and chest, the authors could identify characteristic motion patterns and extract biomechanically descriptive parameters. Furthermore, Ladha et al. [87] present a climbing performance analysis system. They capture a climber's movements through an accelerometer-based wearable sensing platform, automatically detect climbing sessions and moves, and assess parameters related to core climbing skills: power, control, stability and speed.

2.5.2 Healthcare

As discussed in Section 1.2, developing healthcare applications is one of the most important motivations to investigate human activity monitoring. One major goal of such applications is to monitor how far individuals follow recommendations, given in form of either general guidelines or as part of a custom care plan. This information can be used either in the rehabilitation process (for *e.g.* cardiovascular patients) or to promote a more active lifestyle, thus to prevent *e.g.* age-related diseases.

An important factor in healthcare applications related to physical activity monitoring is to motivate the user. By providing online feedback, the user can reflect on his progress and gain insights about his behaviour anywhere and at anytime. This could encourage to continue or do even more physical activity. Related work also investigated how the representation of the results could further improve the user's motivation. For example, using living metaphors [97] or rewarding certain accomplishments with trophies [50] are common motivational tools. In a study conducted by Consolvo et al. [33], users were given an exercise program and a mobile device showing the image of a virtual garden. The users received virtual rewards – *e.g.* flowers appearing in the virtual garden – when performing a certain amount of exercises. The study showed that participants using this system spent significantly more time performing exercises than participants who did not use the system.

Apart from monitoring general physical activity, healthcare applications can also provide valuable information to clinicians to monitor and diagnose certain patients.

For example, the long-term monitoring of a patient's daily life can be used to detect changes or unusual patterns that could indicate early symptoms of diseases such as Alzheimer's or Parkinson's disease. These symptoms might not even occur during short medical appointments. Therefore, integrating results of activity monitoring into out-of-hospital services is of importance. Lau et al. [90] investigate how activity recognition with a smartphone can support patient monitoring and improve telemedicine services.

Finally, a mobile healthcare application called BeWell [27] is shortly described here. It shows how mobile activity monitoring can be integrated into our daily life, and promote multiple aspects of physical and emotional well-being. The BeWell application continuously tracks user behaviour along three distinct health dimensions without requiring any user input. It automatically infers the user's sleep duration (based on phone usage), physical activity (based on the phone's accelerometer, distinguishing between the activity classes *walking*, *running* and *stationary*) and social interaction (based on ambient speech during a day and the usage of social applications on the smartphone). For all three components a score between 0 and 100 is computed. Using these scores, persuasive feedback is given to the user in form of an animated aquatic ecosystem, rendered as an ambient display on the smartphone's home screen [27].

2.5.3 Assisted Living, Elderly Care

A major goal of current research in human activity monitoring is to develop new technologies and applications for the aging. Assisted living is an important application area, with the aim that elderly people live more independent lives. For example, activity recognition systems can assist people suffering from dementia, who tend to forget certain steps while performing an activity. A realization of such a system is presented by Si et al. [161], who developed a prototype of a context-aware reminding system for daily activities. This system helps elderly with dementia to complete different ADLs (*e.g. making tea* or *brushing teeth*), instead of them relying on caregivers. The system is based on a wireless sensor node, which can obtain the information of the elderly person's tool usage in different ADLs. Based on this information, the system provides elderly with a personalized guidance to complete ADLs.

Another important aspect in assisted living is to detect potentially dangerous situations in a person's life and call for external help automatically. Such systems are especially interesting for people living alone, which is the case for many elderly. An example usage is to detect when a person's vital signs indicate imminent health threats, *e.g.* a system to assess heart failure [186]. A further important application area is fall detection, which has been investigated in numerous related work [24, 67, 95, 104, 151].

2.5.4 Industry: Manufacturing and Services

Human activity monitoring has the potential to support workers in their tasks, both in the manufacturing and service sectors. In the context of a large European project called wearIT@work, Lukowicz et al. [102] investigated the use of wearable computing technology for real-life industrial scenarios. Four pilot applications were consid-

ered in the following fields: aircraft maintenance, car production, healthcare, and emergency response. The goal was to use wearable technology and activity recognition to provide a summary of performed activities, to provide hands-free access to *e.g.* electronic manuals, or to assist in the training of new workers.

Concrete examples of the monitoring and support of manufacturing tasks are given *e.g.* in [165, 188]. Ward et al. [188] show an approach of continuous activity recognition using on-body sensing. They combine data from wearable microphones and accelerometers, and recognize a set of workshop activities such as *sawing*, *hammering* or *drilling*. Stiefmeier et al. [165] present a system for tracking workers in car manufacturing plants, investigating two scenarios. In the scenario of an assembly task (the installation of the front lamp) both wearable and environmental sensors are used. This scenario is not feasible in production due to the instrumentation of the cars, thus is restricted to training environments. The second scenario investigates the quality check in the manufacturing process and relies only on wearable sensors integrated into a jacket.

Examples of using human activity recognition for supporting workers in the service sector, concretely in hospital environments are given *e.g.* in [7, 47]. Altakouri et al. [7] investigate to what degree automatic activity recognition could support the use of prioritized lists for mobile phone-based nursing documentation. They show that the activity recognition-based list selection improves both the system's usability and acceptance, considering parameters such as time effort, interaction complexity, error rate and subjective system perception. Finally, Favela et al. [47] demonstrate that mobile activity recognition systems can build pervasive, context- and activity-aware networks for the monitoring of hospital staff, thus providing important information for colleagues.

2.5.5 Other Application Areas

The number of potential application areas of human activity monitoring is numerous. This subsection gives further examples in addition to the above listed areas. The goal of *surveillance applications* is to automatically recognize suspicious behaviour, thus to detect deviations from regular patterns. These systems usually rely on the large number of cameras present in public locations, and face the difficulty of detecting activities from multiple sources. An example is given by Zajdel et al. [198] who detect aggressive human behaviour in public environments, using a fusion of audio and video sensing.

In the application area of *entertainment*, the monitoring and recognition of user's activities has become the focus of interest with the appearance of video games controlled by accelerometers (*e.g.* Nintendo Wii) or even controlled by the player's body (*e.g.* using the Kinect on the Xbox 360 platform). Moreover, many applications of human activity monitoring exist for serious games. An example is presented by Fujiki et al. [59], who monitor the user's performed activities throughout the day. The so calculated activity points are used in a pervasive gaming platform, where players race against each other. Moreover, earned activity points can be spent to get hints in mental games played on the platform, such as Sudoku. The overall goal of the authors

is to encourage physical activity and to modify the user's daily behaviour (*e.g.* taking the stairs instead of the elevator).

In the field of *robotics*, social robots require to detect and track humans and recognize their activities [167]. This is a key aspect to effectively integrate robots into people's workflows, and to natural human-robot interaction in a variety of scenarios. An example of a *military application* is presented by Minnen et al. [111]. They use activity recognition for the automatic generation of post-patrol reports, thus to summarize what happened during a patrol of several hours. Finally, targeted or context-aware *advertising* is an evolving application area. An example is given by Partridge and Begole [121], who display ads that are relevant to the user, based on the user's current or frequent activities.

2.6 Conclusion

In this chapter a general overview of recent, state-of-the-art research related to human activity monitoring has been presented. A wide range of technologies, methods and solutions have been highlighted for the different components and aspects of such systems. Four major topics have been discussed in this chapter, namely the type of monitored activities, the type of applied sensing modalities, different machine learning methods, and finally various application areas of activity monitoring systems. The rest of this thesis will only focus on a fraction of the here presented approaches, which is specified in this section.

From the wide range of activities monitored and recognized in related work, this thesis focuses on low-level activities. The term *physical* or *aerobic activities* will be used throughout this work, referring both to basic locomotion activities (*e.g. walking*, *running* or *cycling*) and further everyday, household and fitness activities. Moreover, the stationary activities (or inactivities) *lying*, *sitting* and *standing* are included, since in many scenarios distinguishing activity and inactivity is important. Overall, the goal is to describe most of an individual's daily routine from the physical activity point of view. The concrete list of included activities will be given in the respective chapters. Generally, this depends on the used dataset, as described in Chapter 3.

Due to the defined list of activities, the usage of ambient sensors is not feasible in this thesis. Therefore, only wearable sensors are considered hereafter. As discussed in Section 4.1, the combination of accelerometers and a heart rate monitor will be used throughout this work. As pointed out in Section 2.3, these types of sensors are complementary to each other. Therefore, as suggested by various related work, the combination of them will be beneficial for recognizing the type and estimating the intensity of performed physical activities.

Considering machine learning methods, Section 4.2 will present a complete data processing chain for physical activity monitoring. This also includes an analysis and evaluation of a wide range of classification methods, focusing thereby only on supervised approaches. Moreover, an important topic of this thesis are meta-level classifiers, especially various boosting algorithms, as presented in Chapter 6.

Finally, as discussed in Section 1.2, the main motivation for developing different methods in this thesis is to be used in healthcare applications. With the precise moni-

toring of physical activities, the here presented solutions can tell how far individuals follow general or custom recommendations. However, the proposed approaches can also be directly used in general fitness applications, where detailed information about the intensity and duration of performed physical activities is of interest for the user.

3

Datasets for Physical Activity Monitoring

3.1 Introduction

Most established research fields are characterized amongst others with publicly available, standard, benchmarked datasets. Such datasets have many benefits: different and new approaches can be compared to each other, no research time has to be spent on laborious data collection, standardized testbeds can be created, etc. In the field of physical activity monitoring ideally datasets reflect natural behaviour, they are recorded from many different subjects performing a wide range of activities, and are fully annotated with ground truth. Unfortunately, due to various difficulties concerning hardware and annotation (all discussed below in this chapter) and due to privacy issues, only a few datasets are publicly available. Moreover, even these few datasets show significant limitations, thus there is a lack of a commonly used, standard dataset. Therefore, this chapter presents two new datasets for physical activity monitoring, both made publicly available for the research community. Moreover, these datasets are used for benchmarking in Chapter 4, showing the difficulty of common classification problems and exposing some challenges in this research field.

3.1.1 Related Work

Recently, datasets for different fields of activity and context recognition have become publicly available. A live-in laboratory was created for the MIT PlaceLab dataset [80]. Volunteers were recorded using devices integrated into the home setting, and cameras installed throughout the house were used for the annotation. Another dataset observing activities performed in a home environment was introduced in [180]. This dataset uses simple binary sensing nodes (*e.g.* reed switches, pressure mats) and particularly contains very long readings (month-long). The dataset presented in [79] focuses on the daily routine of individuals, recording the daily life of one person over a period of 16 days. The Opportunity dataset [103, 144] provides a large recording of 12 subjects performing morning activities (activities of daily living, ADL) in a

room equipped with a kitchen. This dataset uses numerous sensors attached to the body of the participants and in the environment, and contains over 25 hours of sensor data. The TUM Kitchen dataset [174] was created and made publicly available for research in the areas of markerless human motion capture, motion segmentation and human activity recognition. The dataset provides video data from 4 fixed cameras, RFID (radio-frequency identification) tag and reed switch readings and action labels. Finally, Baños et al. [10] presented a benchmark dataset with the specific goal to evaluate sensor displacement in activity recognition. The dataset includes 33 fitness activities, recorded using 9 inertial sensor units from 17 subjects.

The goals of physical (aerobic) activity monitoring are to estimate the intensity of performed activities and to recognize activities like *sitting*, *walking*, *running* or *cycling*. The focus and challenges in this field are – compared to activity recognition in *e.g.* ADL or industrial scenarios – different, due to differing conditions (considering *e.g.* the sensor setup: only a few, wearable sensors can be used). Since the characteristic of the activities in this field also significantly differ from the specific activities of home or industrial settings, different approaches are required, *e.g.* features are calculated usually on longer intervals, etc.

Therefore, datasets specifically created for physical activity monitoring are necessary. However, only a few, limited datasets are publicly available in this research field. The DLR dataset [53] contains 4.5 hours of annotated data from 7 activities performed by 16 subjects, wearing one belt-mounted inertial measurement unit (IMU). Bao and Intille [15] present a data recording of 20 different activities with 20 subjects, wearing five 2-axis accelerometers, and show results in activity recognition with 4 different classifiers. The Opportunity dataset contains 4 basic modes of locomotion: *lying*, *sitting*, *standing* and *walking* [103, 144]. Finally, in the dataset introduced by Xue and Jin [195] a protocol of 10 different activities was followed by 44 subjects, wearing one 3-axis accelerometer.

3.1.2 Problem Statement and Contributions

Data recording for physical activity monitoring faces some difficulties compared to data collection in e.g. home environments, resulting in less comprehensive and established datasets. For instance, a robust hardware setup consisting of only wearable sensors is required. The reason is that activities such as *running* are highly stressing the setup. Moreover, parallel video recording for the purpose of offline annotation – a widely used method in other fields, such as the monitoring of daily activities in home environments – is not feasible if outdoor activities are included in the data collection. Therefore, only online annotation of the performed activities is possible for creating a reliable ground truth. As a result, there is a lack of a commonly used, standard dataset and established benchmarking problems for physical activity monitoring.

This chapter introduces two new datasets for physical activity monitoring, both made publicly available. Based on the conditions and limitations of the public datasets described above, the following criteria were defined for the creation of the datasets: a wide range of everyday, household and fitness activities should be performed by an adequate number of subjects, wearing a few 3D-IMUs and a heart rate

(HR) monitor. The reason for requiring a HR-monitor in addition to the commonly used inertial sensors is that physiological sensing – missing in other public datasets – is especially useful for the intensity estimation of physical activities. For example, inertial sensing alone can not reliably distinguish activities with similar movement characteristic but different energy expenditure, *e.g. walking* and *ascending stairs*, or an even more difficult example: *walking* and *walking with a load*.

A further requirement for the new datasets is that the participating subjects should have the freedom to execute activities however they want. It has been pointed out in previous work (*e.g.* in [15]) that (semi-)naturalistic data collection provides a more realistic training and test data, and permits greater subject variability in behaviour than data recorded in a laboratory setting. For example, subjects should be allowed to freely walk in- or outdoors during data capture, instead of specifying locomotion activities on *e.g.* a treadmill.

The rest of this chapter is organized in the following way: Section 3.2 introduces the PAMAP dataset and Section 3.3 presents the PAMAP2 dataset. The PAMAP dataset contains data from 14 activities and 8 subjects, wearing 3 IMUs and a HR-monitor. The PAMAP2 dataset was recorded with a similar sensor setup from 9 subjects, performing 18 different physical activities. For both datasets, the hardware setup, the data collection protocol, etc. will be described in detail. Moreover, lessons learnt from these data recordings are discussed in Section 3.4. Finally, the chapter concludes in Section 3.5, reflecting also on the impact of making these datasets publicly available.

3.2 The PAMAP Dataset

The PAMAP dataset was recorded with an early system prototype developed in the PAMAP (Physical Activity Monitoring for Aging People) project [116], as part of the aerobic activity monitoring use case. The data collection took place in August 2010. Wired 3D-IMUs and a HR-monitor were used as sensors, and a Sony Vaio ultra-mobile PC (UMPC) as collection unit during data recording. Each of the 8 test subjects followed a predefined data collection protocol of about one hour. Approximately 8 hours of data were collected altogether. The recorded dataset has been made publicly available for research purposes, and can be downloaded from the project's website[1]. This section describes the dataset in more detail, focusing on the hardware setup, participating subjects and the data collection protocol. Supplementary material related to the PAMAP dataset is presented in Appendix B.

3.2.1 Hardware Setup

Inertial data was recorded using 3 wired Colibri inertial measurement units from Trivisio [178]. The sensors are lightweight (22 g without cable) and small ($30 \times 30 \times 13$ mm). Each IMU contains a 3-axis MEMS (micro-electro-mechanical system) accelerometer, a 3-axis MEMS gyroscope, and a 3-axis magneto-inductive magnetic sensor, all sampled at 100 Hz. During data processing in the rest of this thesis, only the

[1] `http://www.pamap.org/demo.html`, entry "PAMAP_Dataset".

Figure 3.1: *PAMAP dataset: placement of IMUs (red dots) and the data collection unit (blue rectangle).*

3-axis accelerometer is used from an IMU, which has a resolution of $0.038\,\mathrm{ms}^{-2}$ in the range ±16g. Of the 3 IMUs, one was attached above the wrist of the dominant arm, one on the chest of the test subjects, and one sensor was foot-mounted.

A Sony Vaio VGN-UX390N UMPC was used as inertial data collection unit, carried by the subjects in a pocket fixed on their belt. The placement of the sensors and this data collection unit is shown in Figure 3.1. The IMUs were connected to the Sony Vaio UMPC by USB-cables, which were taped to the body so that they did not restrict normal movements of the subjects. To obtain heart rate information, the Garmin Forerunner 305, a GPS-enabled sports watch with integrated HR-monitor, was used.

The applied sensors (3 IMUs and a HR-monitor) define 3 positions on a subject's body, since the chest IMU and the HR-monitor are both placed at the same position. Previous work in *e.g.* [122] showed that in the trade-off between classification performance and number of sensors, using 3 sensor locations is the most effective. In systems for physical activity monitoring the number of sensor placements should be kept at a minimum, for reasons of practicability and comfort – since users of such systems usually wear them for many hours a day. On the other hand, a thorough analysis of sensor positions in Section 8.2 shows that less than 3 sensor positions are not sufficient for accurate activity recognition.

During data collection, a supervisor accompanied the test subjects and marked the beginning and end of each of the different activities. These timestamped activity labels were stored on the data collection unit. Synchronization of the timestamped inertial data, annotations and heart rate data was carried out offline. The data format used in the published dataset is given in Appendix B, Table B.1.

Table 3.1: *PAMAP dataset: protocol of data collection. Left side: indoor activities, right side: outdoor activities.*

Activity	Duration [min]	Activity	Duration [min]
Lie	3	Walk very slow	3
Sit	3	Break	1
Stand	3	Normal walk	3
Iron	3	Break	1
Break	1	Nordic walk	3
Vacuum clean	3	Break	1
Break	1	Run	3
Ascend stairs	1	Break	2
Break	2	Cycle	3
Descend stairs	1	Break	1
Break	1	Run	2
Ascend stairs	1	Normal walk	2
Descend stairs	1	Break	2
		Play soccer	3
		Break	2
		Rope jump	2

3.2.2 Subjects

Eight subjects participated in the data collection, seven males and one female. The subjects were employees at a research institute, aged 27.88 ±2.17 years, and had a BMI of 23.68 ±4.13 kgm^{-2}. One subject was left-handed, all the others were right-handed. Detailed information about each of the test subjects is given in Appendix B, Table B.3.

3.2.3 Data Collection Protocol

The protocol of performing indoor and outdoor activities for the data collection is described in Table 3.1, left and right side, respectively. A criterion for selecting activities was on the one hand that the basic activities (*walking*, *running*, *cycling* and *Nordic walking*) and postures (*lying*, *sitting* and *standing*), traditionally used in related work, should be included. On the other hand, everyday (*ascending* and *descending stairs*), household (*ironing*, *vacuuming*) and fitness (*playing soccer*, *rope jumping*) activities were also included to cover a wide range of activities. Moreover, the activity *walk very slow* was introduced to have walking related activities of different intensity levels: *walk very slow – normal walk – run*.

A total of 14 different activities were included in the data collection protocol. The protocol was split into an indoor and an outdoor scenario. The main reason was the limited battery time of the collection unit, but also to avoid the overloading of the test subjects. Each of the subjects had to follow the presented protocol, performing all de-

fined activities in the way most suitable for the subject. Therefore, a semi-naturalistic data collection was carried out when recording the PAMAP dataset, following the specifications defined in Section 3.1.2. A brief description of each of the activities can be found in Appendix B, Table B.5.

One of the goals of physical activity monitoring is to estimate the intensity of performed activities. A HR-monitor is included in the hardware setup of the data collection, this way heart rate related features can be considered for this task during data processing (*cf.* Chapter 4). Therefore, a short break is inserted in the data collection protocol after most of the activities. The duration of the breaks were chosen so that the heart rate of the subjects was allowed to return to the "normal" range after performing an activity. The goal was to ensure that the measured heart rate was unaffected by the previous activities. For this purpose, a 1-minute break was sufficient after most of the activities, except for the most exhausting ones (*ascending stairs*, *running* and *playing soccer*), after which activities a 2-minutes break was inserted. However, since in everyday situations the influence of activities on the next performed ones can not be excluded, this influence was also simulated in the data collection protocol: *descending stairs* was performed directly after *ascending stairs* and *normal walking* directly after *running* (*cf.* Table 3.1).

3.3 The PAMAP2 Dataset

Although the PAMAP dataset provides a good basis to develop and evaluate data processing and classification techniques for physical activity monitoring, a new dataset was recorded for several reasons. First of all, an improved hardware prototype was developed within the PAMAP project [116]. The main advantages are the usage of only wireless sensors and the significantly extended battery time of the collection unit, both clearly making data recording easier. Moreover, the PAMAP dataset includes a significant amount of recordings where data from at least one sensor is missing, thus limiting the applicability of the respective dataset entries. A further goal of a new data collection was to extend the number of activities in the dataset.

Therefore, this section presents the PAMAP2 dataset, recorded in autumn 2011. It includes data from 9 subjects, wearing 3 IMUs and a HR-monitor, and performing 18 different activities. Over 10 hours of data were collected altogether, from which nearly 8 hours were labeled as one of the 18 activities. The dataset has been made publicly available, and can be downloaded from the PAMAP project's website[2]. Moreover, the dataset is included in the UCI machine learning repository [12], named "PAMAP2 Physical Activity Monitoring Data Set" [132].

This section first describes the hardware setup and the subjects participating in the data recording, then the data collection protocol is presented. Supplementary material related to the PAMAP2 dataset is given in Appendix B.

[2] `http://www.pamap.org/demo.html`, entry "PAMAP2_Dataset".

Figure 3.2: *PAMAP2 dataset, GUI used for the data collection: start screen of the labeling tool. This screenshot is made while the subject is performing the activity sit during data collection. All sensors are operating correctly according to the green symbols in the top left corner. Moreover, the subject's heart rate is* 63 *beats per minute in the moment of this screenshot, as indicated in the top left corner as well.*

3.3.1 Hardware Setup

Three IMUs and a heart rate monitor were used as sensors during the data collection. For the inertial measurements, Colibri wireless IMUs from Trivisio [178] were used. The sensors are relatively lightweight (48 g including battery) and small ($56 \times 42 \times 19$ mm). Each IMU contains two 3-axis MEMS accelerometers (range: ± 16 g / ± 6 g, resolution: 13-bit), a 3-axis MEMS gyroscope (range: $\pm 1500°$/s, resolution: 13-bit), and a 3-axis magneto-resistive magnetic sensor (range: $\pm 400\,\mu$T, resolution: 12-bit), all sampled at 100 Hz. To obtain heart rate information, a BM-CS5SR HR-monitor from BM innovations GmbH [21] was used, providing heart rate values with approximately 9 Hz. The sensors were placed onto 3 different body positions. A chest sensor fixation includes one IMU and the heart rate chest strap. The second IMU is attached over the wrist on the dominant arm, and the third IMU on the dominant side's ankle[3], both are fixed with sensor straps.

A Viliv S5 UMPC (Intel Atom Z520 1.33GHz CPU and 1GB of RAM [185]) was used as data collection unit. The main advantage of this device is a battery time of up to 6 hours. A custom bag was made for this collection unit and the 2 USB-dongles additionally required for the wireless data transfer – one for the IMUs and one for the HR-monitor. The bag was carried by the subjects fixed on their belt.

Labeling of the currently performed activities was done via a GUI (graphical user interface) specifically developed for this purpose on the UMPC, *cf.* Figure 3.2 and

[3]It should be noted that, compared to the hardware setup of the PAMAP dataset recording, this sensor is placed on the ankle instead of the subject's foot. This has several reasons: the sensor is easier to mount on the ankle position and is less noticeable. Moreover, the inertial measurements are less dependent from the type of shoes subjects are wearing, thus ensuring a more reliable sensor placement.

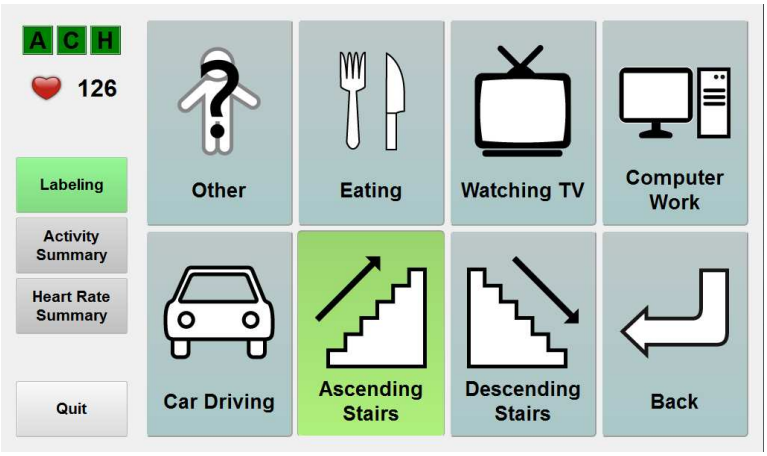


Figure 3.3: *PAMAP2 dataset, GUI used for the data collection: labeling of various everyday activities.*

Figure 3.3. This labeling tool offers the possibility to label the basic activities and postures on its start screen, as shown in Figure 3.2. Moreover, on various other pages, the labeling of a wide range of everyday, household and sport activities is possible as well. Figure 3.3 shows the labeling screen for various everyday activities as an example, while a subject is performing the activity *ascend stairs*. In addition, symbols in the top left corner of the GUI (*cf.* both Figure 3.2 and Figure 3.3) indicate whether the three IMUs and the HR-monitor are continuously operating, thus whether the data acquisition is running smoothly.

During data collection, the beginning and end of each of the different performed activities were marked with the described labeling tool on the Viliv UMPC, thus providing timestamped activity labels along with the raw sensory data. The collection of all raw sensory data and the labeling were implemented in separate threads in an application running on the collection unit, to ease the synchronization of all collected data. The data format of the PAMAP2 dataset is given in Appendix B, Table B.2.

3.3.2 Subjects

In total nine subjects participated in the data collection, eight males and one female. The subjects were mainly employees or students at a research center, aged 27.22 ±3.31 years, and having a BMI of 25.11 ±2.62 kgm^{-2}. One subject was left-handed, all the others were right-handed. Detailed information on each of the test subjects is given in Appendix B, Table B.4.

3.3.3 Data Collection Protocol

A protocol of 12 activities was defined for the data collection, shown in Table 3.2. The criterion for selecting these activities was similar to the explanation given for the PAMAP dataset (*cf.* Section 3.2.3): apart from the basic activities and postures, a wide

Table 3.2: *PAMAP2 dataset: protocol of data collection.*

Activity	Duration [min]	Activity	Duration [min]
Lie	3	Descend stairs	1
Sit	3	Break	2
Stand	3	Normal walk	3
Iron	3	Break	1
Break	1	Nordic walk	3
Vacuum clean	3	Break	1
Break	1	Cycle	3
Ascend stairs	1	Break	1
Break	2	Run	3
Descend stairs	1	Break	2
Break	1	Rope jump	2
Ascend stairs	1		

range of other activities should be included. Each of the subjects had to follow this protocol, performing all defined activities in the way most suitable for them.

Furthermore, a list of optional activities to perform was also suggested to the subjects. The idea of these optional activities was to further enrich the range of activities in the recorded dataset. Activities from this optional list were only performed by some of the subjects if the circumstances made it possible, *e.g.* if the subject had additional free time after completing the protocol, if there was equipment available to be able to perform an optional activity, and if the hardware setup made further data recording possible. In total, 6 different optional activities were performed by some of the subjects: *watching TV*, *computer work*, *car driving*, *folding laundry*, *house cleaning* and *playing soccer*.

The created PAMAP2 dataset therefore contains in total data from 18 different activities. A brief description of each of these activities can be found in Appendix B, Table B.6. Most of the activities from the protocol were performed over approximately 3 minutes, except *ascending/descending stairs* (due to limitations of the building where the indoor activities were carried out) and *rope jumping* (to avoid exhaustion of the subjects). Breaks between activities in the protocol were inserted for the same reason as explained in Section 3.2.3 for the PAMAP dataset. The optional activities were performed as long as the subjects wished, or as long as it took to finish a task (*e.g.* arriving with the car at home or completely finishing dusting a bookshelf).

3.4 Data Collection: Lessons Learnt

This section gives a brief description of lessons learnt during the data capture of the two datasets. The hardware setup used to record the PAMAP dataset was cumbersome. It took over 30 minutes to attach all the sensors and fix the cables. Moreover, the Sony Vaio collection unit's battery time of less than an hour clearly hindered the

data capturing. On the other hand, attaching the sensors and the custom bag for recording the PAMAP2 dataset was straightforward, the entire setup time was not more than 5 minutes. All subjects reported that the sensor fixations were comfortable and did not restrict normal movements at all. Only the custom bag felt sometimes uncomfortable during intensive movements (*e.g. running*). A smaller solution for the collection unit – using *e.g.* a smartphone – would be recommendable for similar data collections.

One aspect, which should not be underestimated, is the weather. Opposed to most of the datasets collected in the research field of activity recognition (recorded *e.g.* in home or industrial settings), a significant part of the two datasets presented in this chapter had to be recorded outdoors. Since most of the subjects preferred not to run or cycle in too hot, cold or rainy conditions, and the entire data collection took several days, careful planning and consulting the weather forecast was required when making the schedule for the subjects.

Problems occurring during such complex and long data recordings are inevitable. The setup belonging to the PAMAP dataset had several weaknesses, due to the wired connection of the IMUs. Overall, this caused a significant amount of data loss, motivating the improvement of the system. As for the setup belonging to the PAMAP2 dataset (using the improved prototype, as described above), there were two main reasons for data loss. The first reason is data dropping caused by glitches in the wireless data transfer. However, this was not too significant: the 3 IMUs had a real sampling frequency (a calculated sampling rate corrected with overall data dropping occurrence) of 99.63 Hz, 99.78 Hz and 99.65 Hz on the hand, chest and ankle placements, respectively (compared to the nominal sampling frequency of 100 Hz). Data loss on the wireless HR-monitor appeared even more rarely, and is also less critical than on the IMUs.

The second, more severe reason for data loss in the PAMAP2 recording was the somewhat fragile system setup due to the additionally required hardware components: 2 USB-dongles, a USB-hub and a USB extension cable were added to the collection unit in the custom bag. Especially during activities like *running* or *rope jumping* the system was exposed to a lot of mechanical stress. This sometimes caused losing connection to the sensors, or even a system crash, when the data recording had to be restarted – and in a few cases the data collection could not be recovered even this way. As a result, some activities for certain subjects are partly or completely missing in the dataset. To try to minimize such problems, it is preferable to use the entire sensor setup from one company (so that no second dongle is needed), or even better would be using sensors with standard wireless communication (although the Trivisio sensors use the 2.4 GHz ISM band, they use a specific communication protocol, and thus a USB-dongle is needed for wireless data streaming). As an alternative, local storage on the sensors should be considered for future data collection, made possible by new sensor solutions recently appearing on the market.

3.5 Conclusion

In the field of physical activity monitoring there is a lack of a commonly used, standard dataset and established benchmarking problems. Therefore, this chapter presented two new datasets (the PAMAP and the PAMAP2 dataset), both made publicly available. The PAMAP dataset was recorded on 14 physical activities with 8 subjects, wearing 3 IMUs and a HR-monitor. The PAMAP2 dataset was recorded with a similar sensor setup from 9 subjects, performing up to 18 different activities. In the respective sections of this chapter the hardware setup, participating subjects and the data collection protocol have been described in detail for both datasets.

Since the introduced new datasets provide a wide range of physical activities, performed by a reasonable number of subjects, challenging classification problems can be defined. This is shown in Chapter 4 where *e.g.* different intensity estimation and activity recognition classification tasks, defined on the PAMAP2 dataset, are benchmarked. The so exposed challenges motivate the improvement of existing data processing and classification approaches, resulting in *e.g.* a new classification algorithm presented in Chapter 6. Moreover, the introduced rich datasets allow the evaluation of everyday life scenarios, and the development of robust techniques and personalization approaches for physical activity monitoring, as shown in Chapter 5 and Chapter 7, respectively.

Apart from using the two introduced datasets in this work, there has been a certain impact in the research community by making them publicly available. The PAMAP dataset was published in October 2011 [139], while the PAMAP2 dataset was published in June 2012 [135, 136]. Moreover, the PAMAP2 dataset was included in the UCI repository in August 2012 [132]. Despite the relatively short time passed since publishing the datasets (this chapter is written in August 2013), several research groups have already made use of them, and state that releasing the datasets is a great service to the research community. Moreover, the number of page hits of the PAMAP2 dataset in the UCI repository [132] passed 11500 (last accessed on 2013-08-29).

Concluding the chapter, this paragraph briefly presents a few major publications from other research fields, which also make use of the PAMAP or PAMAP2 dataset. Rakthanmanon et al. [130] use the PAMAP dataset to demonstrate their multi-dimensional time series clustering algorithm. The PAMAP dataset is used by Hu et al. [71] as one of the examples for real-world problems to evaluate a novel time series classification algorithm under more realistic assumptions. Moreover, Rakthanmanon and Keogh [129] use the PAMAP dataset to evaluate their proposed time series shapelet discovery algorithm, and to demonstrate that shapelets can also be used as a high accuracy classification tool for activity recognition. Huang and Schneider [73] propose spectral learning algorithms for hidden Markov models (HMMs) that incorporate static data, and use the PAMAP2 dataset in their experiments to demonstrate the performance of the new algorithms on real (not synthetic) data. Finally, Clifton et al. [32] introduce an extreme function theory for novelty detection, and illustrate their proposed method on the PAMAP2 dataset, used as a benchmark time-series dataset.

4

Data Processing and Classification

4.1 Introduction

The goals of this thesis in physical activity monitoring are to estimate the intensity of performed activities and to identify basic or recommended activities and postures. These goals are motivated by various health recommendations, as discussed in Chapter 1. For these purposes two datasets have been created: the PAMAP and PAMAP2 datasets, both presented in Chapter 3. These datasets will be used in this chapter to apply different data processing methods and classification algorithms, and to create a benchmark of physical activity classification problems.

The created PAMAP and PAMAP2 datasets provide raw sensory data from 3 inertial measurement units and a heart rate monitor. Previous work shows that for different tasks in physical activity monitoring accelerometers outperform other sensors. Concerning intensity estimation of physical activities, work in *e.g.* [118, 173] show that 3D-accelerometers are the most powerful sensors. In [118] for example, accelerometers and gyroscopes attached to wrist, hip and ankle were used to estimate the intensity of physical activity, and it was found that accelerometers outperform gyroscopes. Concerning the recognition of physical activities, a study carried out by Pärkkä et al. [117] analyzed the effect of various sensors. In this study subjects carried a set of wearable sensors (3D-accelerometer, 3D-compass, microphone, temperature sensor, heart rate monitor, etc.) while performing different activities. According to the results, accelerometers proved to be the most information-rich and most accurate sensors for activity recognition. Pärkkä et al. [117] found that accelerometer signals react fast to activity changes and they reflect the type of activity well. Therefore, from all 3 IMUs, only data from the accelerometers is used hereafter.

In addition to acceleration data, the heart rate signals provided by both datasets will also be used. Physiological signals were closely examined in previous work for intensity estimation. For example, Crouter et al. [36] conclude that combining accelerometer and heart rate data, or using only heart rate information enables good intensity estimation. On the other hand, Tapia et al. [173] show that introducing heart

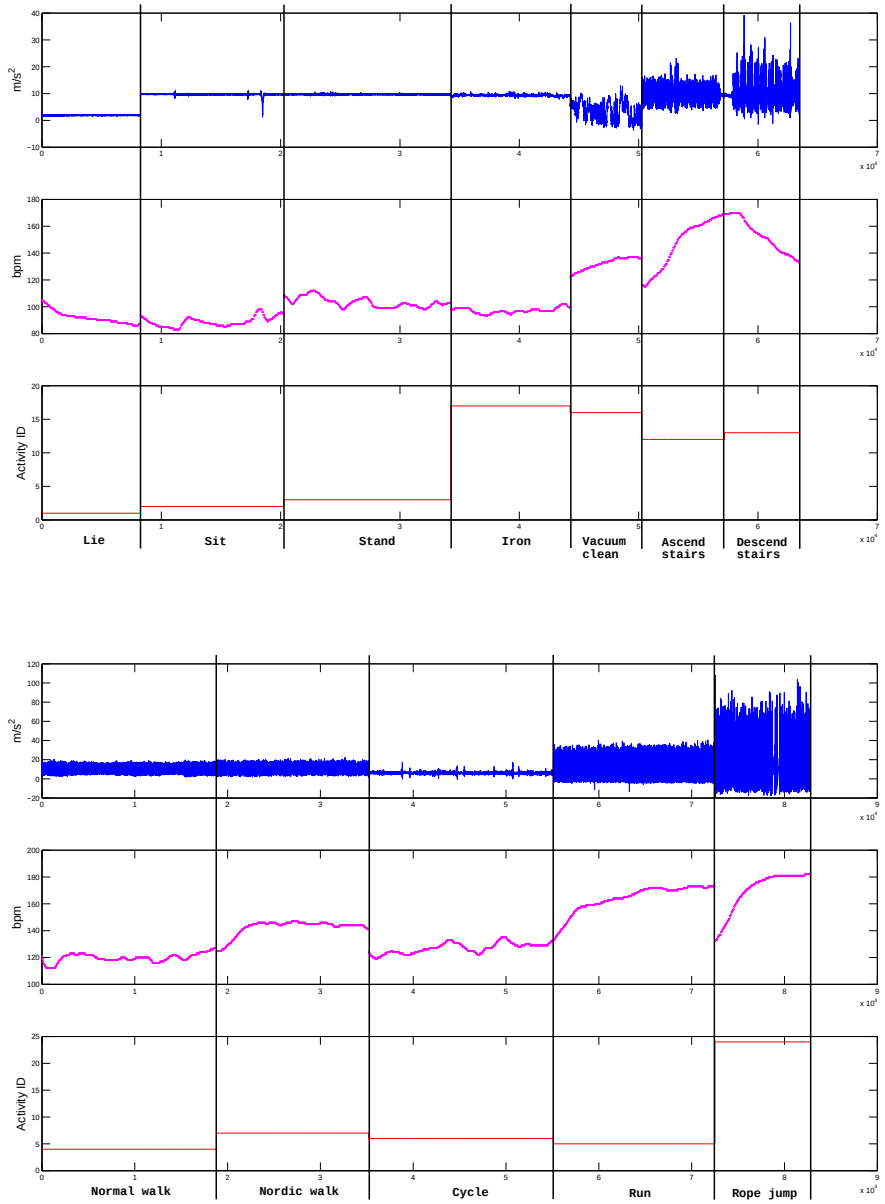


Figure 4.1: *Example raw acceleration (chest IMU up-down direction, shown in the top row of both plots) and heart rate data (shown in the middle row of both plots) from the PAMAP2 dataset (activity IDs as given in the dataset are shown in the bottom row of both plots). The top and bottom plots together show one subject's collected data while performing the* 12 *activities of the defined data collection protocol. Note: data from the break intervals and transient activities has been removed from these plots.*

rate related features has no significant improvement on the differentiation of activities performed with various intensity levels, compared to when only using features derived from acceleration data. Nevertheless, heart rate data will be used in the data processing chain of Section 4.2. Moreover, experiments performed in Section 8.2 will show that heart rate data indeed has the potential to improve the intensity estimates of physical activities.

Figure 4.1 shows raw acceleration (from the chest IMU, up-down direction) and heart rate data from one subject, while performing the various activities defined in the PAMAP2 dataset's data collection protocol. These plots show that different tasks have a different signature, most of them can be easily identified by visual inspection[1]. Therefore, it is clear that with the appropriate data processing steps (*e.g.* the right features extracted or the right classifiers chosen) these differences can be captured, making activity recognition and intensity estimation possible. For example, the posture *lying* can be easily distinguished from the postures *sitting* and *standing* with just the mean value of the presented acceleration signal. Another example is that activities including steps (*walking*-related activities) can be distinguished from other activities (*e.g. cycling* or different postures) by *e.g.* the standard deviation of the up-down direction on the chest or foot/ankle acceleration signal. Moreover, the heart rate signal can be used to distinguish certain activities of differing intensity levels, *e.g. ascending* and *descending stairs*.

This chapter is organized in the following way: Section 4.2 presents how raw sensory data is processed, describing the subsequent processing steps in detail. This data processing chain will be used throughout the rest of this thesis. For the steps preprocessing, segmentation and feature extraction standard, commonly used techniques are applied. The focus of this thesis is on the classification step, for which various algorithms are compared in this chapter and new algorithms will be introduced in the next chapters. Section 4.3 defines the performance measures and evaluation techniques used in the benchmark of Section 4.4. These definitions serve partially as basis for the development of evaluation techniques for robust activity monitoring in Chapter 5. The benchmark presented in Section 4.4 compares commonly used classification algorithms on several tasks, showing their difficulty and exposing challenges in the field of physical activity monitoring. The chapter is summarized in Section 4.5, which also defines some of the main goals addressed in the next chapters.

4.2 Data Processing Chain

This section presents the data processing used in the benchmark in Section 4.4. It can be described as a chain of processing steps, starting from raw sensory data and resulting in a prediction of an intensity and activity class (*cf.* Figure 4.2). It follows a classical approach, which is similar *e.g.* to the activity recognition chain (ARC) presented in [145]. For the first three steps of the here presented data processing

[1] This is true when using raw data from multiple sensors for the comparison. For example, there is little difference in the acceleration measured on the chest IMU while performing *normal walking* and *Nordic walking* (*cf.* Figure 4.1). However, the difference is clearly noticeable between these two activities when comparing acceleration measured on the arm IMU

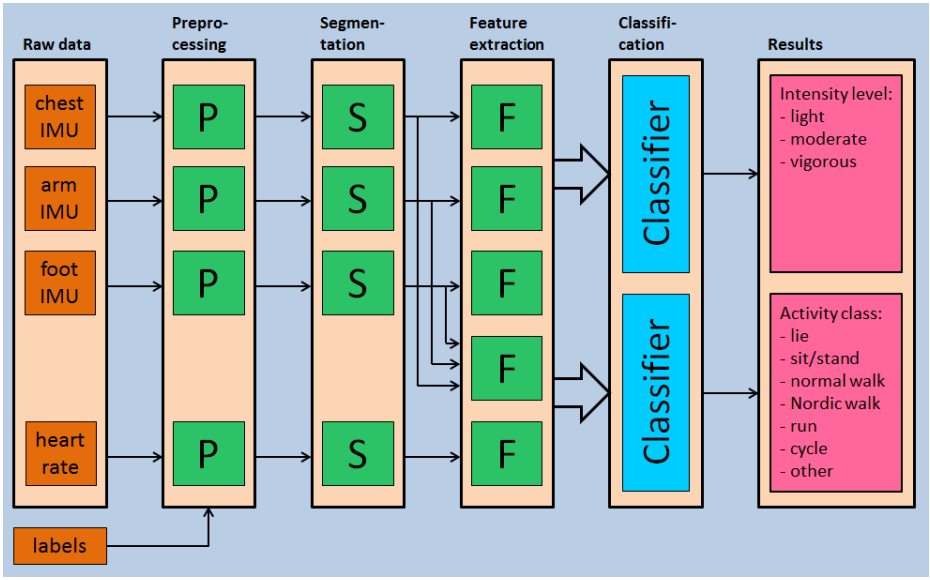


Figure 4.2: *The data processing chain. From raw sensory data synchronized, timestamped and labeled 3D-acceleration and heart rate data is obtained during the preprocessing step (**P**). This data is segmented in the segmentation step (**S**) with a sliding window, using a window size of* 512 *samples. A total of 137 different features are extracted in the next, the feature extraction step (**F**). These features serve as input to the classifiers, which output the estimated intensity level class and the recognized activity class.*

chain (DPC) – preprocessing, segmentation and feature extraction – common choices are made and justified in the next subsections. As for the classification step, various classifiers are introduced and compared, highlighting their benefits and drawbacks.

Compared to the ARC of [145], the decision making consists of only one step: a classifier module takes the entire feature set as input, and outputs a class of the given classification problem. Another example of a similar DPC for physical activity recognition was presented by Baños et al. [11]. They argue to perform sensor fusion at the classification level (sensor fusion is performed at the feature extraction level in the DPC applied in this chapter). The benefit is that when sensors are added or removed, the system does not require a complete retraining. The challenges of the approach proposed by Baños et al. [11] are robustness and scalability, thus the system has to be accurate enough independent of the topology or the number of sensors considered.

In this thesis, since both activity recognition and intensity estimation are regarded as classification problems (*cf.* Section 4.4.1), the same data processing steps can be applied on both of them. The goal in this chapter is not aiming for the best performance on the defined classification tasks, but to provide a baseline characterization with the benchmark. The results presented here and the challenges exposed serve as motivation to improve existing methods, as shown in Chapter 6 with the introduction of a new boosting algorithm, and in Chapter 7 with the development of novel personalization approaches.

Table 4.1: *Data processing: list of extracted features.*

Time domain features	Frequency domain features
Mean	Energy
Median	Entropy
Standard deviation	Dominant frequency
Peak of absolute data	Power ratio of certain frequency bands
Absolute integral	
Correlation between axes	
Gradient	

4.2.1 Preprocessing

The previously introduced datasets provide timestamped raw sensory data from the 3 IMUs and the heart rate monitor, and timestamped activity labels. All this data is synchronized in the preprocessing step. After this step synchronized, timestamped and labeled acceleration (as justified above, only data from the accelerometers is used from all IMUs) and heart rate data is available.

To deal with wireless data loss (thus handling missing values when applying data processing and classification techniques), Saar-Tsechansky and Provost [148] proposed different methods. Linear interpolation was selected from these approaches for simplicity reasons. Further processing of the raw signals (*e.g.* filtering) is included in the extraction of various features, as described in Section 4.2.3. Finally, to avoid dealing with potential transient activities, 10 seconds from the beginning and the end, respectively, of each labeled activity is deleted.

4.2.2 Segmentation

Previous work shows (*e.g.* [75]) that for segmentation there is no single best window length for all activities. To obtain at least two or three periods of all different periodic movements, a window length of about 3 to 5 seconds is reasonable. For example, experiments presented by Lara et al. [89] showed best results with a window size of 5 seconds when using acceleration data for physical activity recognition. Therefore, and to assure effective discrete Fourier transform (DFT) computation for the frequency domain features[2], a window size of 512 samples was selected. Since the sampling rate of the raw sensory data was 100 Hz in both the PAMAP and the PAMAP2 datasets, the segmentation step results in signal windows of 5.12 seconds length. Therefore, the preprocessed data is segmented using a sliding window with the defined 5.12 seconds of window size, shifted by 1 second between consecutive windows.

[2]Compare *e.g.* the commonly used Cooley-Tukey fast Fourier transform (FFT) algorithm [34], which recursively breaks down a DFT of a discrete signal of length N into smaller DFTs. This procedure is the most computationally effective when N is a power of 2.

4.2.3 Feature Extraction

From the segmented 3D-acceleration data, various signal features are computed in both time and frequency domain (*cf.* [48] for an overview and classification of extracted features from sensory data). In addition to the most commonly used features in related work (mean, median, standard deviation, peak acceleration and energy), some other features – also proved to be useful in previous work – are computed, too. The absolute integral feature was successfully used to estimate the metabolic equivalent in *e.g.* [118]. Correlation between each pair of axes is especially useful for differentiating among activities that involve translation in just one or multiple dimensions, *e.g.* walking, running vs. ascending stairs [131]. Power ratio of the frequency bands 0–2.75 Hz and 0–5 Hz proved to be useful in [134]. Peak frequency of the power spectral density (PSD) was used for the detection of cyclic activities in *e.g.* [42]. Spectral entropy of the normalized PSD is a useful feature for differentiating between locomotion activities (walking, running) and cycling [42]. A list of all extracted features is given in Table 4.1.

For a mathematical definition of the extracted features the following notation is used. After the segmentation step of the DPC a discrete-time sequence of N elements is given ($N = 512$). This can be represented as an $N \times 1$ vector: $\boldsymbol{x} = [x_0, x_1, \ldots, x_{N-1}]^T$, where x_i refers to the ith element of the data sequence. The mean value of this vector is computed as follows:

$$\text{mean}(\boldsymbol{x}) = \mu_{\boldsymbol{x}} = \frac{1}{N} \sum_{i=0}^{N-1} x_i. \tag{4.1}$$

The median value is defined as follows: After arranging all the $x_0, x_1, \ldots, x_{N-1}$ data from the lowest to the highest value, the median is the middle element. The standard deviation of the data vector is defined as:

$$\text{std}(\boldsymbol{x}) = \sigma_{\boldsymbol{x}} = \sqrt{\frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu_{\boldsymbol{x}})^2}. \tag{4.2}$$

The peak of the absolute data is defined as follows:

$$MAX_{abs}(\boldsymbol{x}) = \max(|\boldsymbol{x}|). \tag{4.3}$$

The feature absolute integral on the discrete-time sequence is defined as:

$$INT_{abs}(\boldsymbol{x}) = \sum_{i=0}^{N-1} |x_i|. \tag{4.4}$$

Finally, correlation between one pair of axes is defined as follows, assuming the two discrete-time sequences are denoted as $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$r_{\boldsymbol{xy}} = \frac{\text{cov}(\boldsymbol{x}, \boldsymbol{y})}{\sigma_{\boldsymbol{x}} \sigma_{\boldsymbol{y}}}. \tag{4.5}$$

For the frequency-domain features first the power spectral density is computed: $X_0, X_1, \ldots, X_{N-1}$, where X_k refers to the kth element of the PSD. The feature energy is defined then as following:

$$\text{Energy} = \sum_{k=0}^{N-1} X_k^2. \tag{4.6}$$

The feature spectral entropy is defined as:

$$\text{Entropy} = -\sum_{k=0}^{N-1} X_k \log X_k. \tag{4.7}$$

The dominant frequency of the PSD is defined as follows:

$$f_{max} = f_m \quad m = \arg\max_k X_k. \tag{4.8}$$

Finally, the power ratio of two frequency bands which consist of the first p and q elements of the PSD, respectively, is defined as follows:

$$\text{Power\_ratio}_{p,q} = \frac{\sum_{k=0}^{p-1} X_k}{\sum_{k=0}^{q-1} X_k}. \tag{4.9}$$

The signal features extracted from the 3D-acceleration data are computed for each axis separately, and for the 3 axes together, too. This results in 108 (= 9 types of features ×3 sensors ×4, since all 3 axes and their combination is calculated) plus 9 (for the feature correlation between each pair of axes, calculated for all 3 sensors) extracted features. Moreover, since synchronized data from the 3 IMUs is available, combining sensors of different placements is possible. From the above mentioned features (and calculated on 3 axes of each of the IMUs) mean, standard deviation, absolute integral and energy are pairwise (*e.g.* arm plus chest sensor placement) weighted accumulated. Furthermore, a weighted sum of all the 3 sensors together is also added[3]. This combination of different sensors results in 16 additionally extracted features. From these derived features it is expected that they would better describe and distinguish activities with *e.g.* both upper and lower body movement, thus improving the recognition of activities involving movements of multiple body parts. Moreover, considering especially the features containing all 3 sensor placements, these features could improve the intensity estimation of activities. Overall, 133 (= 108 + 9 + 16) features are extracted from the segmented IMU acceleration data.

From the heart rate data, the features (normalized) mean frequency and the frequency gradient are calculated. Normalization is done on the interval defined by resting and maximum HR. The resting HR of a test subject is extracted from the 3 minutes *lying* task in the data collection protocol (*cf.* Section 3.2.3 and Section 3.3.3 for the PAMAP and PAMAP2 datasets, respectively), and is defined as the lowest HR

[3]The weights were selected heuristically, and are set to 0.5, 0.2 and 0.3 for the chest, arm and foot/ankle sensor locations, respectively. The goal was to receive more meaningful features compared to when simply accumulating the feature values from the different sensor placements.

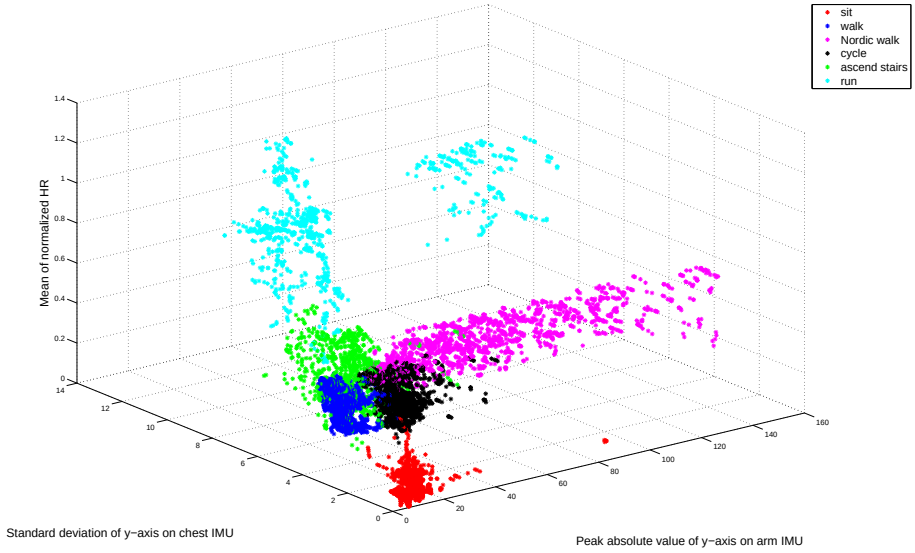


Figure 4.3: *Data processing: example visualization of the feature space. The* 6 *selected physical activities can mostly be distinguished with the* 3 *chosen features (computed from acceleration and heart rate data).*

value measured over this period. As for the maximum HR (MHR), the subject's age-predicted MHR ($MHR = 220 - age$) is used [173]. The feature gradient, both on the raw and normalized heart rate signal, is defined as the difference between the first and last element of a window segment:

$$\text{grad}(\mathbf{x}) = x_{N-1} - x_0. \tag{4.10}$$

Overall, 4 features are extracted from the segmented heart rate data. Therefore, in total 137 (= 133+4) features are derived from each of the 5.12 seconds long signal segments, yielding a large feature vector. Different techniques have been applied in the field of physical activity monitoring to reduce the feature space, *e.g.* principal component analysis (PCA) [8, 192] or Sammon's mapping [38]. Moreover, various feature selection methods were applied in other previous work. For example, distribution bar graphs were created in [117], a heuristic greedy forward search was performed in [187] and a forward-backward sequential search algorithm was applied in [124] to select the best features. However, in this thesis, no feature selection or reduction of the feature space is applied on the extracted feature set. The reason is that the focus of this thesis is on the classification step of the DPC, thus all features will be used for each of the classifiers presented hereafter.

Figure 4.3 visualizes a part of the feature space: samples of 6 selected activities are shown using 3 selected features. The purpose of this plot is to show that differentiating between the performed physical activities is feasible with the set of extracted features. For example, due to larger arm movements, samples of *walk* and *Nordic walk* can mostly be separated with *e.g.* the feature peak absolute value, computed on the

forward-backward direction on the arm accelerometer. Another example can be observed when using the mean value of the normalized heart rate: This feature is very useful when distinguishing between *ascend stairs* and *walk* or *Nordic walk*. Depending on the complexity of the classification task (thus *e.g.* the number of activities to be distinguished) a larger set of features might be required. The selection of these features and how they are used to separate different classes is realized by the respective classification algorithms, which are presented in the next subsection.

4.2.4 Classification

The extracted features serve as input for the next processing step, the classification. In the field of physical activity monitoring research, especially activity recognition, different classification approaches exist and yielded good results. The benefit of using the data processing chain of Figure 4.2 is amongst others its modularity. This allows to easily remove any module and replace it with a different approach, thus different classifiers can easily be tested and compared to each other. This subsection presents various classification algorithms which are commonly used in related work. Preliminary studies are carried out, using data provided by the PAMAP dataset. Based on the results of these studies, several classifiers are selected for the benchmarking process, as described in Section 4.4.2.

In this thesis, both intensity estimation and activity recognition are regarded as classification problems. For the preliminary studies of this subsection, one classification task is defined for each, which are justified and described in more detail in Section 4.4.1. For the intensity estimation task 3 classes are defined: The goal is to distinguish activities of light, moderate and vigorous effort. From the 14 physical activities included in the PAMAP dataset lying, sitting, standing, ironing and very slow walking are regarded as activities of light effort; vacuum cleaning, descending stairs, normal walking, Nordic walking and cycling as activities of moderate effort; ascending stairs, running, playing soccer and rope jumping as activities of vigorous effort. For the activity recognition task 7 classes are defined: lying, sitting/standing (forming one class), normal walking, Nordic walking, running, cycling and other. The latter class includes all remaining activities from the PAMAP dataset[4]: ironing, vacuum cleaning, ascending and descending stairs, playing soccer and rope jumping.

For the training and evaluation of all classifiers within the preliminary studies of this subsection (except of the custom decision tree classifier), the Weka toolkit is used [65]. Weka (Waikato Environment for Knowledge Analysis) is a free machine learning software written in Java. It provides tools for analyzing and understanding data, including the implementation of a large amount of data mining algorithms, and a graphical user interface for easy data manipulation and visualization. A great description of the Weka toolkit can be found in [193], along with a thorough grounding in the machine learning concepts the toolkit uses, and practical advice for using the different tools and algorithms.

[4]Except of very slow walking since this activity was only included in the dataset for the intensity estimation task, in order to have walking related activities in all 3 intensity classes.

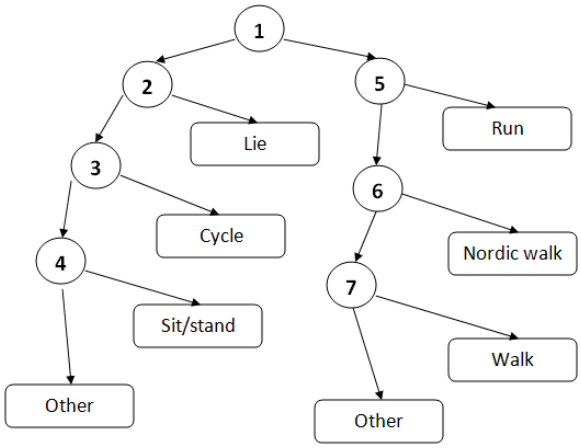


Figure 4.4: *Structure of the custom decision tree classifier, created for the activity recognition task (PAMAP dataset).*

The various classification schemes and results of the preliminary studies are presented in the following paragraphs.

Custom decision tree

Custom decision tree classifiers have been successfully applied to activity recognition in numerous previous work [15, 17, 43, 83, 117]. The advantages of custom decision trees include low computational requirements, simple implementation and a good understanding of the classifier's structure. Figure 4.4 shows the structure of the custom decision tree constructed for the above defined activity recognition task. The tree has 7 binary decision nodes and 8 leaf nodes, the latter representing the different activities. The first decision node divides all activities into activities with and without footsteps, all other decisions are used to separate one activity from the remaining other activities. If the current sample is not recognized into any of the activities while passing the decision tree, it falls through to the default *other* class. Features for the decision nodes were selected using the cluster precisions method, introduced in [75]. More details about applying custom decision tree classifiers for physical activity monitoring problems can be found in [133].

Although custom decision trees have some benefits, the results in [133] indicate that for more complex classification problems (*e.g.* including an increased number of activities) more advanced classification techniques are required. The main drawback of custom decision trees is that they do not necessarily provide the most suitable classifier for a given set of features. In order to exhaustively explore the solution space, other classifiers (*e.g.* automatically generated decision trees) have to be investigated.

Base-level classifiers

Base-level classifiers have been widely used for activity monitoring classification tasks. For example, automatically generated decision trees are applied in [15, 17, 61, 117], k-

Table 4.2: *Accuracy on the activity recognition task with 4 different base-level classifiers, applied on the PAMAP dataset.*

Classifier	**Accuracy [%]**
C4.5	85.03
kNN	87.62
SVM	62.31
Naive Bayes	74.14

nearest neighbors (kNN) in [61, 108], support vector machines (SVM) in [39], Naive Bayes classifiers in [61, 100], or artificial neural networks (ANN) in [43, 63, 117]. A comparison of base-level classifiers for activity recognition can be found *e.g.* in [8, 122]. From the different classification approaches, C4.5 decision tree, kNN, SVM and Naive Bayes classifiers are tested in a preliminary study performed on the activity recognition task, defined above on the PAMAP dataset. Each of these 4 classification methods were named as one of the top 10 data mining algorithms, identified by the IEEE International Conference on Data Mining (ICDM) in December 2006 [194].

A detailed introduction into decision tree classification can be found in [147]. C4.5 is a widely used algorithm to generate decision tree classifiers and is implemented in the Weka toolkit. A practical description of choices and settable parameters of this algorithm is given in [193].

The k-nearest neighbor algorithm (originally proposed by Fix and Hodges [51]) belongs to the instance-based learning methods. In kNN, a new feature vector is classified based on the k closest training examples in the feature space.

Support vector machine classifiers select a small number of critical boundary instances called *support vectors* from each class, and build a linear discriminant function that separates them as widely as possible [193]. SVM is a useful and popular classification technique not only because it constructs a maximum margin separator, but also because – by using different kernel functions – it is possible to form nonlinear decision boundaries with it. The Weka toolkit uses the *libsvm* library, practical advice for using this tool can be found in [70].

Finally, the Naive Bayes classifier is a simple probabilistic classifier, probably the most common Bayesian network model used in machine learning. The model assumes that the features are conditionally independent of each other, given the class. The Naive Bayes model works surprisingly well in practice, even when the conditional independence assumption is not true. A great overview on probabilistic learning, Bayesian classifiers and learning Bayesian models is given in[147].

The preliminary study with these 4 classification methods was carried out with the Weka toolkit [65]. For evaluation, leave-one-subject-out 8-fold cross-validation protocol was used (more details on this subject independent evaluation technique can be found in Section 4.3.2). The accuracy (performance measures are described in more detail in Section 4.3.1) of the different base-level classifiers is shown in Table 4.2, overall good results were achieved. However, the results also indicate a possible further improvement on classification accuracy, since even the best result was only 87.62%. Therefore, there is a reasonable demand for developing more complex

and more advanced classifiers to obtain better results in activity monitoring classification tasks.

Meta-level classifiers

The use of meta-level classifiers for physical activity monitoring problems (*cf. e.g.* [205]) is not as widespread as using various base-level classifiers. However, the comparison of base-level and meta-level classifiers on different activity recognition classification tasks in [131] showed that meta-level classifiers (such as boosting, bagging, plurality voting, etc.) outperform base-level classifiers, thus applying them is of interest. Detailed information on ensemble learning (meta-level classifiers) can be found *e.g.* in [147, 193].

From the various meta-learning algorithms provided by the Weka toolkit, two ensemble learning methods are selected and evaluated in this chapter: bagging and boosting. The idea behind these methods is to iteratively learn weak classifiers by manipulating the training dataset, and then combining the weak classifiers into a final strong classifier. To briefly describe bagging and boosting, assume that the training dataset contains N instances: $(\underline{x}_i, y_i)$ $i = 1, \ldots, N$ ($\underline{x}_i$ is the feature vector, y_i is the annotated class of the instance: $y_i \in 1, \ldots, C$), and $t = 1, \ldots, T$ iterations are performed with the weak classifier $f(\underline{x})$. In bagging, N instances are randomly sampled with replacement in each t iteration from the instances of the training dataset. The learning algorithm (the $f(\underline{x})$ weak classifier) is applied on this sample, the resulting model is stored. After learning, when classifying a new instance, a class is predicted with each of the T stored models. The final decision is the class that has been predicted most often.

The difference in boosting is that the training dataset is reweighted after each iteration, and the single learning models are also weighted for constructing the final strong classifier. This way, the weak learners built in the subsequent iterations focus on classifying the difficult instances correctly. Moreover, when constructing the final classifier, more influence is given to the more successful models. There exist many variants based on the idea of boosting, *cf.* Chapter 6 for a thorough description of them. One of the most widely used variants is AdaBoost, which is also implemented in the Weka toolkit. Moreover, AdaBoost is identified as one of the top 10 data mining algorithms by Wu et al. [194], thus it will be used by the further experiments of this chapter.

Both bagging and boosting use the same learning algorithm (the same type of weak classifier, *e.g.* a decision tree classifier) in each iteration, and combine these T models into the final strong classifier. In a preliminary study all 4 above tested base-level classifiers (C4.5 decision tree, kNN, SVM and Naive Bayes) are evaluated as learning algorithms for both bagging and boosting, using the Weka toolkit. On the kNN classifier (which performed best in the experiments performed above, *cf.* Table 4.2) no improvement was observed applying neither boosting nor bagging. This observation is in accordance with the results of [131]. On the other hand, boosting and bagging the C4.5 classifier resulted in a significant improvement of classification accuracy. Moreover, from all the base-level and meta-level classifiers tested within this subsection, best results were achieved with boosted decision trees.

Table 4.3: *Confusion matrix on the intensity estimation task, performed on the PAMAP dataset. The results were achieved with an AdaBoost C4.5 classifier. The table shows how the intensity class of different annotated samples is estimated in [%].*

Annotated intensity	**Estimated intensity**		
	light	moderate	vigorous
light	96.49	3.15	0
moderate	5.24	89.74	5.02
vigorous	0	2.32	97.68

Table 4.4: *Confusion matrix on the activity recognition task, performed on the PAMAP dataset. The results were achieved with an AdaBoost C4.5 classifier. The table shows how different annotated activities are classified in [%].*

Annotated activity	**Recognized activity**						
	1	2	3	4	5	6	0
1 lie	100	0	0	0	0	0	0
2 sit/stand	0	82.75	0.56	0	0	0.26	16.42
3 walk	0	0	87.05	0.42	3.54	0	8.99
4 Nordic walk	0	0	1.16	79.22	5.14	0	14.48
5 run	0	0	0	0	96.71	0	3.29
6 cycle	0	3.10	0	0	0	82.94	13.96
0 other	0	4.37	0.04	0.04	0.13	0.08	95.35

This paragraph gives more detailed results from the preliminary experiments, achieved with the best performing classifier (AdaBoost with C4.5 decision trees) on the PAMAP dataset. Table 4.3 shows the confusion matrix on the intensity estimation task. The overall accuracy using leave-one-subject-out 8-fold cross-validation is 94.37% with the boosted decision tree classifier. It is worth mentioning that misclassifications only appear into "neighbour" intensity classes, thus no samples annotated as light intensity were classified into the vigorous intensity class, and vice versa. Table 4.4 shows the confusion matrix on the activity recognition task, defined on the PAMAP dataset. The overall accuracy using leave-one-subject-out 8-fold cross-validation is 90.65% with the boosted decision tree classifier. Most of the misclassifications can be explained with the introduction of the *other*, background activity class: The characteristics of some of the other activities overlap with some of the basic activity classes to be recognized. For example, the activities *standing* and *ironing* or *running* and *playing soccer* have similar characteristics. The problem of dealing with background activities for activity recognition is further analyzed in Chapter 5. There, methods will be proposed and evaluated in order to develop robust activity monitoring systems for everyday life.

Table 4.5: *General confusion matrix of a classification task.*

Annotated class	**Recognized class** 1	2	...	C	
1	$P_{1,1}$	$P_{1,2}$	...	$P_{1,C}$	S_1
2	$P_{2,1}$	$P_{2,2}$	...	$P_{2,C}$	S_2
⋮					
C	$P_{C,1}$	$P_{C,2}$	...	$P_{C,C}$	S_C
	R_1	R_2	...	R_C	

4.3 Performance Evaluation

This section introduces and describes the methods used for evaluation in the benchmark. This includes the definition of performance measures and evaluation techniques applied.

4.3.1 Performance Measures

For physical activity monitoring, data is usually collected following a given protocol, as presented during the recording of the PAMAP and PAMAP2 datasets in Chapter 3. It is common practice to delete the beginning and the end of each labeled activity (10 seconds are deleted in the presented preprocessing step of the DPC, *cf.* Section 4.2.1). Therefore, contrary to *e.g.* activity recognition in home or industrial settings, the ground truth is much less fragmented, and there is less variability in event (activity) length. For continuous activity recognition, new error metrics were introduced recently, *e.g.* insertion, deletion, merge, fragmentation, overfill, etc. [181, 189]. However, the goals of physical activity monitoring – as justified above – are usually restricted to frame by frame recognition (thus not the events are important, but the time spent performing each of the activities). Therefore, the frame by frame evaluation methods describe the performance of the used classifiers well, and are regarded as sufficient for benchmarking in this chapter.

The commonly used performance measures are applied for creating the benchmark: precision, recall, F-measure and accuracy. For the definition of these metrics *cf.* the notation in the general confusion matrix in Table 4.5. The performance measures are defined generally and will be used for different classification problems (activity recognition and intensity estimation tasks) in the benchmark of Section 4.4.

Assume that a confusion matrix is given by its entries $P_{i,j}$, where i refers to the rows (annotated classes), and j to the columns (recognized classes) of the matrix (*cf.* Table 4.5). Let S_i be the sum of all entries in the row i of the matrix (referring to the number of samples annotated as class i), and R_j the sum of all entries in the column j of the matrix (referring to the number of samples recognized as class j). Let N be the total number of samples in the confusion matrix. Let the classification problem represented in the confusion matrix have C classes: $1, 2, \ldots C$. Using this notation, the performance measures precision and recall are defined as follows:

$$precision = \frac{1}{C} \sum_{i=1}^{C} \frac{P_{i,i}}{R_i} \tag{4.11a}$$

$$recall = \frac{1}{C} \sum_{i=1}^{C} \frac{P_{i,i}}{S_i}. \tag{4.11b}$$

Therefore, precision can be interpreted as a measure of exactness (how reliable the results are in a class), while recall can be interpreted as a measure of completeness (how complete the results are of a class). Considering both the precision and the recall, F-measure is traditionally defined as the harmonic mean of them:

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{4.11c}$$

Finally, the measure accuracy is defined as the percentage of correctly classified samples out of all samples:

$$accuracy = \frac{1}{N} \sum_{i=1}^{C} P_{i,i}. \tag{4.11d}$$

It should be noted that, from the above defined metrics, accuracy only considers the total number of samples. As for the other metrics, class imbalance is taken into account: normalization is done using the total number of samples for each class separately. This different behaviour of the performance measures is important since fewer samples from some activities in a dataset are not necessarily due to lesser importance of these activities, but could be caused by *e.g.* a more difficult data capture of these activities. For example, the created PAMAP and PAMAP2 datasets are also characterized by being imbalanced, thus certain activities occur more frequently than others (*cf.* Chapter 3). Some results in Section 4.4.3 will also point out the difference between the performance metrics, and how these results should be interpreted.

4.3.2 Evaluation Techniques

A commonly used evaluation technique to validate machine learning methods is k-fold cross-validation (CV). This technique randomly partitions a dataset into k equal size subsets. From these k subsets, $k-1$ are used as training data and the remaining subset as test data. This procedure is repeated k times, so that each of the k subsets are used exactly once as test data.

In the field of physical activity monitoring, one of the evaluation goals is to simulate how the developed methods apply to a new user. Therefore, to simulate subject independency, the evaluation technique leave-one-subject-out (LOSO) CV is applied. Since benchmarking is performed on the PAMAP2 dataset which provides data from 9 subjects, LOSO 9-fold CV is performed. This means that data from 8 subjects is used for training and data from the remaining subject is used for testing, repeating this procedure 9 times leaving always another subject's data for testing.

Section 5.1.2 will reflect on the differences between subject dependent and independent evaluation, concluding that for physical activity monitoring systems usually subject independent validation techniques should be applied. However, in order to create a widely used and comparable benchmark, both subject dependent and subject independent evaluation is carried out. As for the subject dependent evaluation, standard 9-fold cross-validation is applied in the benchmark ($k = 9$ was chosen to have the same number of folds as for the subject independent evaluation).

4.4 Benchmark of Physical Activity Monitoring

Benchmarked classification problems are important for a research field for various reasons, *e.g.* to show the difficulty of different tasks or to show where the challenges lie. However, the field of physical activity monitoring lacks established benchmarking problems. This is due to the fact that only a few datasets are publicly available in this area, as discussed in Chapter 3. Bao and Intille [15] use their recorded dataset to show results with 4 different classifiers on an activity recognition task. The Opportunity dataset [103, 144] contains 4 basic modes of locomotion, on which a recognition task is defined and included in the benchmark of [149]. Finally, Xue and Jin [195] present a benchmark on an activity recognition task defined on their created dataset. They use an SVM classifier with different sets of features.

The above listed existing benchmarks of activity monitoring problems are very limited, concerning the number of physical activities and sensors included in the dataset, and concerning the applied classification algorithms. Moreover, they only define one activity recognition task each, thus classification problems with different complexity and an intensity estimation task are not included. Therefore, to overcome the limitations of existing benchmarks, this section presents benchmarking on the PAMAP2 physical activity monitoring dataset. First different classification tasks are defined, then a set of classifiers is selected. The benchmark is created with the standard data processing chain, as presented in Section 4.2. The performance measures and evaluation techniques, as defined in Section 4.3, are applied for the benchmark. Overall, 4 classification tasks are benchmarked with 5 different classifiers. The benchmark results are shown and discussed in Section 4.4.3.

4.4.1 Definition of the Classification Problems

The definition of classification tasks for the benchmark follows two goals. First of all, both activity recognition and intensity estimation tasks should be included to cover the main objectives of physical activity monitoring. Furthermore, classification tasks of different complexity should be included. Therefore, 4 classification problems are defined in total for benchmarking, which are described below. These tasks only focus on the 12 activities performed during the data collection protocol of the PAMAP2 dataset, the 6 optional activities are left out from this benchmark.

Intensity estimation task

Intensity estimation is regarded as a classification task throughout this chapter. Three

classes are defined for this problem: activities of light, moderate and vigorous effort. Obtaining ground truth for this task is less straightforward than for activity recognition tasks, hence requires a short explanation.

In various previous works on estimating intensity of physical activity (*e.g.* in [35, 118]), reference data was collected with a portable cardiopulmonary system (*e.g.* Cortex Metamax 3B or Cosmed K4b^2). This method has the advantage that it provides precise measurements on an individual's oxygen consumption. It makes measured metabolic equivalents (METs) available, which is essential if the task is to use these values to *e.g.* estimate metabolic equivalent from other features [35, 118]. However, in this thesis the goal is to only estimate whether a performed activity is of light, moderate or vigorous effort, since for the physical activity recommendations only this information is needed (*cf.* Section 1.2). Therefore, it is sufficient to use the Compendium of Physical Activities [1] to obtain reference data for the defined intensity estimation task. This compendium contains MET levels assigned to 605 activities. It was *e.g.* used in the recommendations given by Haskell et al. [66] to provide example activities of moderate and vigorous intensities. Moreover, the compendium was used for validation of MET estimation in related work, *e.g.* in [109].

The ground truth for the defined rough intensity estimation task is thus based on the metabolic equivalent of the different activities, provided by Ainsworth et al. [1]. Therefore, the 3 classes are defined as following using the set of activities from the PAMAP2 dataset: lying, sitting, standing and ironing are regarded as activities of light effort (< 3.0 METs); vacuum cleaning, descending stairs, normal walking, Nordic walking and cycling as activities of moderate effort (3.0-6.0 METs); ascending stairs, running and rope jumping as activities of vigorous effort (> 6.0 METs).

Basic activity recognition task

Five activity classes are defined for this problem: lying, sitting/standing, walking, running and cycling. All other activities are discarded for this task. This classification problem refers to the many existent activity recognition applications only including these, or a similar set of few basic activities. The ground truth for this task – and for the other two activity recognition tasks presented below – is provided by the labels made during data collection. The activities sitting and standing are forming one class in this problem. This is a common restriction made in activity recognition (*e.g.* in [43, 120]), since an extra IMU on the thigh would be needed for a reliable differentiation of these postures. The numerous misclassifications between these two postures appearing in the results belonging to task 'all' in the benchmark (*cf.* Section 4.4.3) confirm that these two activities can not be reliably distinguished with the given set of sensors.

Background activity recognition task

Six classes are defined for this problem: lying, sitting/standing, walking, running, cycling and other. The latter class consists of the remaining 6 activities of the PAMAP2 data collection protocol: ironing, vacuum cleaning, ascending stairs, descending stairs, Nordic walking and rope jumping. The idea behind the definition of this task is that

in physical activity monitoring users always perform meaningful activities. However, there are countless number of activities, and – apart from a few, for the particular application relevant activities – an exact recognition is not needed. On the other hand, ignoring these other activities would limit the applicability of the system. Therefore, the introduction of a background activity class is justified. Section 5.1.1 will reflect more on the introduction of other activities, describing the concept of a background activity class in detail.

All activity recognition task

Twelve activity classes are defined for this problem, corresponding to the 12 activities of the data collection protocol.

4.4.2 Selected Classifiers

This subsection presents and justifies the selection of a set of classifiers used in the benchmarking process. Section 4.2.4 introduced various classifiers for physical activity monitoring, preliminary studies were carried out with them on the PAMAP dataset. Based on these experiments, the best performing base-level and meta-level classifiers are selected for the benchmark. From the base-level classifiers, the C4.5 decision tree, Naive Bayes and kNN are selected, based on the results of Table 4.2. Moreover, since boosting and bagging of the C4.5 classifier showed the most significant improvement in the experiments with meta-level classifiers, these ensemble learners are selected as well for creating the benchmark.

Therefore, 5 different classifiers are selected, a thorough comparison of these classification techniques will be given in the benchmark results of Section 4.4.3 on the PAMAP2 dataset. Each of these classification approaches are frequently used in related work. Overall they represent a wide range of classifier complexity. The experiments for creating the benchmark are all performed with the Weka toolkit [65]. The five classifiers are listed below, together with the parameters differing from the default values set in the Weka toolkit. These parameters were determined heuristically and used successfully in the preliminary studies in Section 4.2.4 . Moreover, for reproducibility and easier comparability with future results, the exact definition (scheme) of each of the classifiers – as given in the Weka toolkit – is included in the following list of the five classifiers:

1. Decision tree (C4.5)
   - confidenceFactor = 0.15
   - minNumObj = 50
   - *Scheme:weka.classifiers.trees.J48 -C 0.15 -M 50*

2. Boosted C4.5 decision tree
   - confidenceFactor = 0.15 (in the decision tree)
   - minNumObj = 50 (in the decision tree)
   - *Scheme:weka.classifiers.meta.AdaBoostM1 -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 – -C 0.15 -M 50*

3. Bagging C4.5 decision tree
   - confidenceFactor = 0.15 (in the decision tree)
   - minNumObj = 50 (in the decision tree)
   - *Scheme:weka.classifiers.meta.Bagging -P 100 -S 1 -I 10 -W weka.classifiers.trees.J48 – -C 0.15 -M 50*

4. Naive Bayes
   - *Scheme:weka.classifiers.bayes.NaiveBayes*

5. kNN
   - KNN = 7 (number of neighbours)
   - *Scheme:weka.classifiers.lazy.IBk -K 7 -W 0*

4.4.3 Results and Discussion

This subsection presents and discusses the benchmark on the 4 classification tasks (*cf.* Section 4.4.1), performed with the 5 selected classifiers (*cf.* Section 4.4.2). Both subject dependent and subject independent evaluation (*cf.* Section 4.3.2) results are shown for all classifier/classification problem combinations. Tables 4.6 – 4.9 present the results in form of the 4 defined performance measures (*cf.* Section 4.3.1). In the rest of this subsection, some conclusions are drawn and discussed.

Overall, the best performance was achieved with the kNN and the boosted decision tree classifiers. This confirms the results of the preliminary studies in Section 4.2.4. Concerning the Naive Bayes classifier, it is interesting to observe how it performs on the different tasks among the evaluated base-level classifiers. On classification problems having clear class boundaries (the tasks 'basic' and 'all') it performs better than the decision tree classifier. On the other hand, the decision tree classifier outperforms the Naive Bayes classifier on the other two problems (the classification tasks 'intensity' and 'background'): These tasks have classes containing multiple activities, thus it is difficult to define the class boundaries with the Naive Bayes classifier – contrary to the decision tree classifier.

Another general observation can be made when comparing the results of subject dependent and independent evaluation: The former indicates highly "optimistic" performance. Due to this significant performance difference, usually subject independent validation techniques should be preferred for physical activity monitoring, in order to present results with practical meaning. This issue will be discussed in more detail in Chapter 5.

Results on the different classification tasks are generally in accordance with previous observations. For instance, the best classifiers not only achieve approximately 96% on the intensity estimation task, but misclassifications only appear into "neighbour" intensity classes, as already observed in Table 4.3. Concerning the 'background' task, all performance measures significantly decreased compared to the 'basic' task, *e.g.* from 97.94% to 94.33% with the boosted decision tree classifier. The reason why the complexity of the classification problem increased so significantly was briefly discussed with the results of Table 4.4: The characteristics of some of the introduced

Table 4.6: *Benchmark on the PAMAP2 dataset: performance measures on the 'Intensity estimation task'.*

	Standard 9-fold cross-validation				LOSO 9-fold cross-validation			
Classifier	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure	Accuracy
C4.5 decision tree	0.9796	0.9783	0.9789	0.9823	0.9490	0.9364	0.9426	0.9526
Boosted C4.5	0.9989	0.9983	0.9986	0.9988	0.9472	0.9564	0.9518	0.9587
Bagging C4.5	0.9853	0.9809	0.9831	0.9866	0.9591	0.9372	0.9480	0.9552
Naive Bayes	0.9157	0.8553	0.8845	0.9310	0.8986	0.8526	0.8750	0.9251
kNN	0.9985	0.9987	0.9986	0.9982	0.9488	0.9724	0.9604	0.9666

Table 4.7: *Benchmark on the PAMAP2 dataset: performance measures on the 'Basic activity recognition task'.*

	Standard 9-fold cross-validation				LOSO 9-fold cross-validation			
Classifier	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure	Accuracy
C4.5 decision tree	0.9968	0.9968	0.9968	0.9970	0.9349	0.9454	0.9401	0.9447
Boosted C4.5	0.9997	0.9994	0.9995	0.9995	0.9764	0.9825	0.9794	0.9785
Bagging C4.5	0.9971	0.9968	0.9970	0.9971	0.9346	0.9439	0.9392	0.9433
Naive Bayes	0.9899	0.9943	0.9921	0.9923	0.9670	0.9737	0.9703	0.9705
kNN	1.0000	1.0000	1.0000	1.0000	0.9955	0.9922	0.9938	0.9932

Table 4.8: *Benchmark on the PAMAP2 dataset: performance measures on the 'Background activity recognition task'.*

	Standard 9-fold cross-validation				LOSO 9-fold cross-validation			
Classifier	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure	Accuracy
C4.5 decision tree	0.9784	0.9701	0.9743	0.9709	0.8905	0.8635	0.8768	0.8722
Boosted C4.5	0.9991	0.9979	0.9985	0.9980	0.9559	0.9310	0.9433	0.9377
Bagging C4.5	0.9881	0.9766	0.9823	0.9787	0.9160	0.8937	0.9047	0.9042
Naive Bayes	0.8905	0.9314	0.9105	0.8508	0.8818	0.8931	0.8874	0.8308
kNN	0.9982	0.9966	0.9974	0.9957	0.9428	0.9458	0.9443	0.9264

Table 4.9: *Benchmark on the PAMAP2 dataset: performance measures on the 'All activity recognition task'.*

	Standard 9-fold cross-validation				LOSO 9-fold cross-validation			
Classifier	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure	Accuracy
C4.5 decision tree	0.9554	0.9563	0.9558	0.9546	0.8376	0.8226	0.8300	0.8244
Boosted C4.5	0.9974	0.9973	0.9974	0.9969	0.8908	0.8947	0.8928	0.8796
Bagging C4.5	0.9660	0.9674	0.9667	0.9666	0.8625	0.8489	0.8556	0.8554
Naive Bayes	0.9419	0.9519	0.9469	0.9438	0.8172	0.8561	0.8362	0.8365
kNN	0.9946	0.9937	0.9942	0.9925	0.9123	0.9097	0.9110	0.8924

background activities overlap with some of the basic activity classes to be recognized. This issue will be further investigated in Chapter 5.

Altogether, good performance is achieved on all 4 classification tasks: approximately 90% or more with the best performing classifiers. However, there are two important challenges defined by the benchmark, where more advanced approaches in future work should improve the performance. On the one hand, by increasing the number of activities to be recognized – while keeping the same sensor set – the difficulty of the task exceeds the potential of standard methods. This not only applies for the task 'all', but for the 'background' task as well: By introducing an *other* activity class for all the background activities, the complexity of the classification problem significantly increases, thus the performance drops using the same standard approaches. On the other hand, when comparing classification performance individually for the 9 subjects, a high variance can be observed. This strongly increases with the increase of task complexity: The individual performance on the 'basic' task (using the boosted decision tree classifier) varies between 93.99% and 100%, while on the 'all' task it varies between 74.02% and 100%. Therefore, especially on the more difficult classification problems, personalization approaches (subject dependent training) could significantly improve compared to the results of the benchmark.

4.5 Conclusion

This chapter presented data processing methods and classification algorithms for physical activity monitoring. A data processing chain is defined including preprocessing, segmentation, feature extraction and classification steps. For the first three steps common approaches are used in this thesis. For the classification step, different algorithms are introduced and compared. First preliminary studies are carried out with a wide range of classifiers using the PAMAP dataset. Moreover, a benchmark is given in this chapter by applying 5 selected classifiers on 4 defined classification tasks.

The presented results mainly serve to characterize the difficulty of the different tasks. The benchmark reveals some challenges in physical activity monitoring, which will be addressed in the next chapters. For example it shows that complex activity recognition tasks exceed the potential of existing approaches. This motivates the introduction of new classification algorithms, as presented in Chapter 6. Moreover, the large variance of individual classification performance motivates novel personalization approaches, as discussed in Chapter 7.

The definition and benchmark of classification problems including the 6 optional activities from the PAMAP2 dataset remains for future work. Furthermore, it should be noted that a post-processing step is not included in the DPC as defined in this chapter. Therefore, no temporal information is taken into account when classifying activities. The reason is that when following a protocol during data collection, there is no practical meaning how different activities follow each other. However, in real-life situations patterns in the order of performed activities exist: For example driving car is usually preceded and followed by walking and not *e.g.* by sitting or especially not *e.g.* by ironing clothes. To simulate this, datasets recorded directly from subjects'

everyday life have to be created. Then, methods such as HMMs can be applied for determining the transition between different types of physical activities. However, this problem exceeds the purpose of this thesis.

5

Robust Activity Monitoring for Everyday Life: Methods and Evaluation

5.1 Introduction

In literature, the monitoring of physical activities under realistic, everyday life conditions – thus while an individual follows his regular daily routine – is usually neglected or even completely ignored. Therefore, this chapter investigates the development and evaluation of robust methods for everyday life scenarios, with focus on the tasks of aerobic activity recognition and intensity estimation. Two important aspects of robustness are investigated: dealing with various (unknown) other activities and subject independency, both explained in more detail in the next subsections. Methods to handle these issues are proposed and compared. The usage of activity monitoring applications in common everyday scenarios is thoroughly evaluated in simulations. Moreover, a new evaluation technique is introduced (leave-one-activity-out, LOAO) to simulate when an activity monitoring system is used while performing a previously unknown activity. Through applying the proposed methods it is possible to design a robust physical activity monitoring system with the desired generalization characteristic.

The outline of this chapter is the following: the current section describes the problem statement related to the other activities and subject independence. Section 5.2 defines the basic conditions (classification problems, data processing and classification methods) of the experiments carried out in this chapter. Section 5.3 proposes four different models for dealing with the other activities in the activity recognition classification task. The measures used to quantify the classification performance of the different approaches are defined in Section 5.4, adjusted to the focus of the activity recognition and intensity estimation tasks. Section 5.5 presents the evaluation techniques used in the experiments in this chapter. Results on each of the defined classification tasks are presented and discussed in Section 5.6. A detailed analysis of the results is supported by various confusion matrices achieved with different combinations of classifier, other activity model and evaluation technique. Finally, the developed methods and obtained results are summarized in Section 5.7.

5.1.1 Problem Statement: Other Activities

The recognition of basic aerobic activities (such as walk, run or cycle) and basic postures (lie, sit, stand) is well researched, and is possible with just one 3D-accelerometer [42, 100]. However, since these approaches only consider a limited set of similar activities, they only apply to specific scenarios. Therefore, current research in the area of physical activity recognition focuses amongst others on increasing the number of activities to recognize. For example, 11 different activities are recognized in [122], 16 different activities of daily living (ADL) in [78], 19 different activities (with focus on locomotion and sport activities) in [8], and 20 different everyday activities are distinguished in [15], etc. However, there are countless different activities (*e.g.* 605 different activities are listed in [1]), thus it is not feasible to recognize all of them – not only due to the highly increased complexity of the classification problem, but also due to the fact that collecting data from those hundreds of different activities is practically not possible.

In practice, activity monitoring systems usually focus on only a few activities of interest. Therefore, the main goal is to recognize only these few activities, but as part of a classification problem where all other activities are included as well. Thus the other activities do not need to be recognized, but should not be completely ignored either. One possible way to handle uninteresting other activities is to add a null-class rejection stage at the end of the activity recognition chain, thus discard instances of classified activities based on the confidence of the classification result [145]. Another possibility is to handle them as sub-activities clustered into the main, basic activity classes, *e.g.* ascend/descend stairs considered as walk [100]. The drawback of this solution is that there still remain many activities which can not be put into any of the basic activity classes (*e.g.* vacuum clean or rope jump). The concept of a null-class (also called background activity class) has been successfully used in the field of activity spotting, *e.g.* in [115]. This concept will be further investigated for aerobic activity recognition in this chapter: Apart from the few activities to be recognized, all other activities are part of this null activity class in the defined problem. This inclusion of the other activities increases the applicability of the system, but also significantly increases the complexity of the classification problem, as shown by different experiments performed within this chapter.

Example 5.1

Here a practical use case is given where introducing and dealing with other activities would be beneficial. The authors of [196] present an approach for energy-efficient continuous activity recognition on mobile phones by introducing the 'A3R' (Adaptive Accelerometer-based Activity Recognition) strategy. In A3R, both the accelerometer sampling frequency and the choice of the classification features are adapted in real-time, based on the currently performed activity. However, the A3R strategy goes into an unknown state when not confident enough in the estimation of the activity class. In this unknown state, the energy consumption is the highest (maximum sample rate and using all features). Yan et al. [196] noted that in the in-site Android study users performed many other activities beyond the 6 labeled ones, causing the appearance of the unknown state more frequently, thus resulting in higher energy consumption.

With an additional *other* activity class, covering a large number of usually performed other activities and assigning an adequate sampling rate and feature set to it, the unknown state would occur less frequently, thus reducing the overall energy consumption of the mobile application.

The above mentioned approaches represent a first important step towards dealing with various other activities. However, they only handle a given set of other activities (the entire set of other activities is known when developing the system), thus neglect to simulate the – in practice important – scenario when the user of the system performs an activity previously unknown to the system. Therefore, it remains an open question what happens to all the activities not considered during the monitoring system's development. To give a concrete example, assume that an activity monitoring system has the goal to recognize 5 basic physical activities (walk, run, etc). When developing this system, in addition to the activities to recognize, 10 other activities are considered as well (vacuum clean, play soccer, etc). The system is specified so that if a user performs any of these other activities, it is not recognized as a basic activity but as an other activity or is rejected. Furthermore, assume that the activity 'rope jump' is neither included in the basic, nor in the set of other activities. Therefore, it is undefined how the system handles the situation when a user performs this 'rope jump' activity. By not dealing with this issue, existing approaches leave basically two possibilities: either the user is limited to scenarios where only the considered activities occur (even if 20 – 30 different activities are included in the development of a system, this still is a significant limitation for the user), or the user is permitted to perform any kind of physical activity, but it is not specified how the monitoring system handles an activity not considered during the system's development phase (*e.g.* whether it is recognized as one of the basic activities). Either way, by neglecting this issue, the applicability of an activity monitoring application is significantly limited.

5.1.2 Problem Statement: Subject Independency

Another important aspect of robustness is the subject independency of an activity monitoring system. In [122], a comparison of subject dependent and independent validation is shown, and a large difference of classifier performance is reported between the two validation techniques (1.26 – 5.92% misclassification *vs.* 12.09 – 29.47% misclassification for different classifiers, respectively). Moreover, the benchmark in Section 4.4 also compared subject dependent and independent evaluation, pointing out the significant performance difference between the two methods. Overall, for physical activity monitoring – unless the development of personalized approaches is the explicit goal – subject independent validation techniques should be preferred. This best simulates the common scenario that such systems are usually trained on a large number of subjects and then used by a new subject (similar to the concept of unknown other activities as discussed above, here the user of the system is unknown during the development phase). In contrast, subject dependent evaluation leads to too "optimistic" performance results. However, still many recent research works use subject dependent validation techniques (*e.g.* in [89, 177]). Hence, neglecting that

although they present high performance using their approach, these results might not have as much practical meaning as if subject independent validation would have been applied. Further results comparing subject dependent and independent evaluation techniques will be shown in this chapter.

5.2 Basic Conditions of the Experiments

This section defines the 3 classification problems used within this chapter: the 'basic', 'extended' and 'intensity' tasks. The main motivation to define these classification tasks is to cover the two essential goals of physical activity monitoring, namely activity recognition and intensity estimation. Moreover, this section specifies the data processing and classification methods applied to solve the introduced classification problems.

5.2.1 Definition of the Classification Problems

The experiments performed within this work are all based on the PAMAP2 dataset, a physical activity monitoring dataset created and released recently [136, 135], and included in the UCI machine learning repository [12]. This dataset is used since it not only includes the basic physical activities (walk, run, cycle, Nordic walk) and postures (lie, sit, stand), but also a wide range of everyday (ascend and descend stairs, watch TV, computer work, drive car), household (iron clothes, vacuum clean, fold laundry, clean house) and fitness activities (rope jump, play soccer). The dataset was recorded from overall 18 physical activities performed by 9 subjects, wearing 3 inertial measurement units (IMU) and a heart rate monitor. A more detailed description of the dataset can be found in Section 3.3.

The overall goal of activity recognition in this chapter is to develop a physical activity monitoring system which can recognize a few, basic activities and postures of interest, but is also robust in everyday situations. In their daily routine, users of activity monitoring systems perform a large amount of different activities, many of them are not of interest from the activity recognition point of view. Therefore, to simulate this common usage of activity monitoring systems, the activity recognition task is defined as follows.[1] There are 6 different basic activity classes to recognize: lie, sit/stand[2], walk, run, cycle and Nordic walk. In addition, 9 different activities are regarded as other/background activities: iron clothes, vacuum clean, ascend stairs, descend stairs, rope jump, fold laundry, clean house, play soccer and drive car. These other activities should not be recognized as one of the basic activities, but as part of an other activity class or should be rejected. The additional activities will be used to simulate the scenario when users perform other activities than the few basic ones,

[1] The defined classification task uses 16 different activities from the PAMAP2 dataset. The remaining 2 activities (computer work and watch TV) are discarded here due to their high resemblance to the basic postures.

[2] It is a common restriction made in activity recognition (*e.g.* in [43]) that the postures sit and stand form one activity class, since an extra IMU on the thigh would be needed for a reliable differentiation of them.

and are also used to simulate the scenario when users perform an activity unknown to the system. This defined classification problem will be referred to as 'extended' activity recognition task throughout this chapter. Moreover, for comparison reasons, the classification problem only including the 6 basic activity classes will also be used, and will be referred to as 'basic' activity recognition task.

The defined activity recognition tasks focus on the monitoring of traditionally recommended aerobic activities (*walk*, *run*, *cycle* and *Nordic walk*), and can thus be justified by various physical activity recommendations – as given in [66]. Especially patients with diabetes, obesity or cardiovascular disease are often required to follow a well defined exercise routine as part of their treatment. Therefore, the recognition of these basic physical activities is essential to monitor the progress of the patients and give feedback to their caregiver. Moreover, a summary of resting activities (*lie*, *sit* and *stand still*) also gives feedback on how much sedentary activity the patients "performed". However, in other use cases the focus of an activity monitoring application could be different, thus the definition of the classification problem (the definition of the basic and other activity classes) would differ. Nevertheless, the methods presented in this chapter could be applied to those other classification tasks as well.

Apart from the activity recognition tasks presented above, an intensity estimation classification task is also defined on the PAMAP2 dataset. This task will be referred to as the 'intensity' classification problem, and will be used to demonstrate the necessity of subject independent evaluation and to simulate the estimation of the intensity of previously unknown activities. The 'intensity' task includes all 18 activities from the PAMAP2 dataset, the goal is to distinguish activities of light, moderate and vigorous effort. The ground truth for this rough intensity estimation task is based on the metabolic equivalent (MET) of the different physical activities, provided by [1]. Therefore, the 3 intensity classes are defined as follows: lie, sit, stand, watch TV, computer work, drive car, iron, fold laundry and clean house are regarded as activities of light effort (< 3.0 METs); walk, cycle, Nordic walk, descend stairs and vacuum clean as activities of moderate effort (3.0-6.0 METs); run, ascend stairs, play soccer and rope jump as activities of vigorous effort (> 6.0 METs). Overall, the 'intensity' task is regarded as a 3-class classification problem in this chapter.

5.2.2 Data Processing and Classification

The PAMAP2 dataset provides raw sensory data from the 3 IMUs and the heart rate monitor, which needs to be first processed in order to be used by classification algorithms. A data processing chain is applied on the raw data including preprocessing, segmentation and feature extraction steps (these data processing steps are further described in Section 4.2). In total, 137 features are extracted from the raw signal: 133 features from IMU acceleration data (such as mean, standard deviation, energy, entropy, correlation, etc.) and 4 features from heart rate data (mean and gradient). These extracted features serve as input for the classification step, together with the activity class labels provided by the dataset.

Previous work in physical activity monitoring showed that decision tree based classifiers, especially boosted decision trees, usually achieve high performance (*cf.*

e.g. [137] or the benchmark results in Section 4.4). Moreover, decision tree based classifiers have the benefit to be fast classification algorithms with a simple structure, and are thus also easy to implement. These benefits are especially important for physical activity monitoring applications since they are usually running on mobile, portable systems for everyday usage, thus the available computational power is limited. Therefore, the C4.5 decision tree classifier [126] and the AdaBoost.M1 (using C4.5 decision tree as weak learner) algorithm [55] are used and compared in the experiments on the defined classification problems.

5.3 Modeling Other Activities

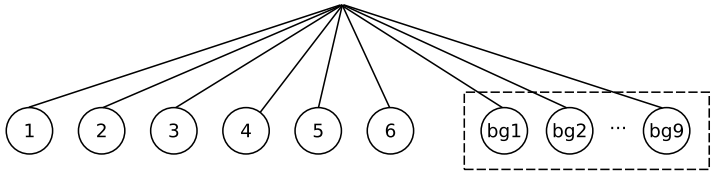
As discussed above, the focus of the activity recognition tasks is on the recognition of the basic activity classes, but all the other activities should not be completely neglected either. Therefore, 4 different models are proposed for dealing with these other activities. The main goal of these solutions is a high recognition rate of the basic activities, and also robust behaviour concerning unknown activities, thus having good generalization characteristic. The 4 proposed methods are listed below, and are visualized – by means of the concrete example of the defined 'extended' activity recognition task: 6 basic activity classes and 9 other activities – in Figure 5.1.

- **The 'allSeparate' model:** for each of the other (or also called background) activities a separate activity class is defined ('bg1' ... 'bg9'), and all these classes are regarded as activities not belonging to the 6 basic activity classes ('1' ... '6'). The concept of the method is visualized in Figure 5.1a. This model refers to the nowadays common approach of dealing with a large number of activities: most research work is focused on increasing the number of recognized activities, thus to have a high number of separate activity classes.

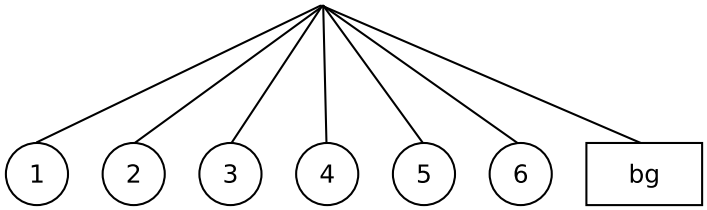
- **The 'bgClass' model:** in addition to the basic activity classes a background activity class ('bg' in Figure 5.1b) is defined, containing all the other activities. This approach of a null-class for physical activity recognition was proposed in [137] to increase the applicability in everyday life scenarios.

- **The 'preReject' model:** it basically inserts a null class rejection step before the actual classification. The concept of this two-level model is visualized in Figure 5.1c. On the first level the basic activities are separated from all the other activities ('bg' class). The second level – only on the 'basic' branch of the first level – distinguishes the 6 different basic activity classes. When constructing a classifier based on this model, all training samples are used to create the subclassifier of the first level, while for the second level only training samples from the basic activity classes are used.

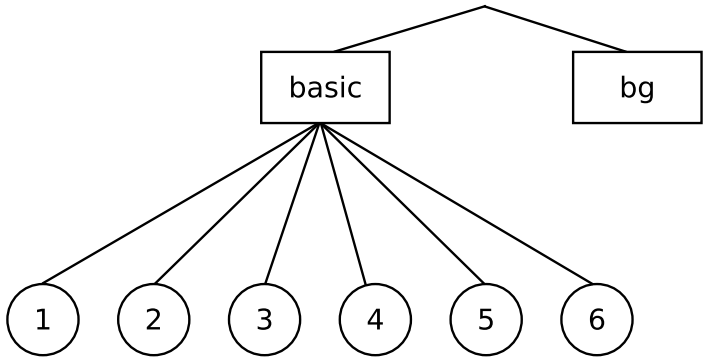
- **The 'postReject' model:** similar to the 'preReject' model, this is also a two-level model, as shown in Figure 5.1d. However, the null class rejection step is applied after classifying the basic activities. This solution is similar to *e.g.* the decision filtering step applied after activity classification in the activity recognition chain of [145]. Only samples from the basic activity classes are used to



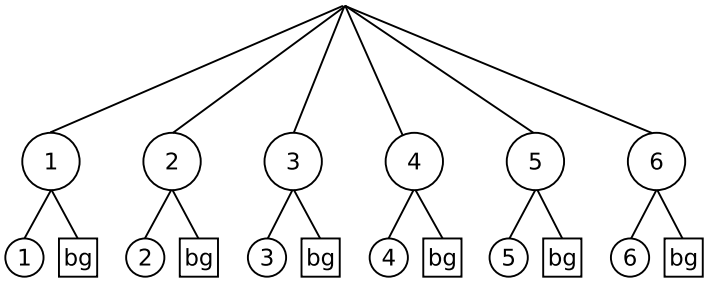
(a) *The 'allSeparate' model.*



(b) *The 'bgClass' model.*



(c) *The 'preReject' model.*



(d) *The 'postReject' model.*

Figure 5.1: *The 4 proposed models for dealing with the other activities.*

create the first level of this classifier, while the second level consists of 6 sub-classifiers: each created using the respective basic activity class and all samples from the other activities.

5.4 Performance Measures

The common performance measures, as derived from the general confusion matrix in Section 4.3.1, are used to quantify the classification performance of the different approaches: precision, recall, F-measure and accuracy. However, these measures are adjusted to the specific focus of the defined activity recognition and intensity estimation classification tasks, as shown in Section 5.4.1 and Section 5.4.2, respectively.

Table 5.1: *Confusion matrix used for the adjusted definition of the performance measures for the activity recognition tasks.*

Annotated activity	**Recognized activity** 1	2	...	C	0	
1	$P_{1,1}$	$P_{1,2}$	...	$P_{1,C}$	$P_{1,C+1}$	S_1
2	$P_{2,1}$	$P_{2,2}$	...	$P_{2,C}$	$P_{2,C+1}$	S_2
...						
C	$P_{C,1}$	$P_{C,2}$	...	$P_{C,C}$	$P_{C,C+1}$	S_C
$C+1$	$P_{C+1,1}$	$P_{C+1,2}$	...	$P_{C+1,C}$	$P_{C+1,C+1}$	S_{C+1}
...						
$C+B$	$P_{C+B,1}$	$P_{C+B,2}$	...	$P_{C+B,C}$	$P_{C+B,C+1}$	S_{C+B}
	R_1	R_2	...	R_C	R_{C+1}	

5.4.1 Activity Recognition

The focus of the activity recognition classification tasks in this chapter is on the recognition of the basic activity classes, thus the performance measures are adjusted accordingly. This adjusted definition of the 4 measures uses the following notation (*cf.* also the confusion matrix in Table 5.1). Assume that a confusion matrix is given by its entries $P_{i,j}$, where i refers to the rows (annotated activities), and j to the columns (recognized activities) of the matrix. Let S_i be the sum of all entries in the row i of the matrix (referring to the number of samples annotated as activity i), and R_j the sum of all entries in the column j of the matrix (referring to the number of samples recognized as activity j). Let N be the total number of samples in the confusion matrix. Let the classification problem represented in the confusion matrix have C basic activity classes: $1,\ldots,C$ and B other activity classes: $1,\ldots,B$. Let the activity classes ordered so in the confusion matrix that the background activity classes follow the basic activity classes (*cf.* the order of the annotated activity classes in Table 5.1). Since the classification of the samples belonging to the other activities is not of interest, this is represented as a null activity class in the confusion matrix (*cf.* the column referred to as $P_{i,C+1}$ in Table 5.1). Samples classified as one of the background activity classes

Table 5.2: *General confusion matrix of the intensity estimation task, using annotated intensity classes.*

Annotated intensity	**Estimated intensity** light	moderate	vigorous	
light	$P_{1,1}$	$P_{1,2}$	$P_{1,3}$	S_1
moderate	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$	S_2
vigorous	$P_{3,1}$	$P_{3,2}$	$P_{3,3}$	S_3
	R_1	R_2	R_3	

('allSeparate' model), or classified into the other activity class ('bgClass' model), or rejected before or after the classification of the basic activities ('preReject' or 'postReject' model, respectively) are counted into this null class. Using this notation, the performance measures precision and recall are defined as following:

$$precision = \frac{1}{C} \sum_{i=1}^{C} \frac{P_{i,i}}{R_i} \tag{5.1a}$$

$$recall = \frac{1}{C} \sum_{i=1}^{C} \frac{P_{i,i}}{S_i}. \tag{5.1b}$$

For F-measure, the original definition as presented in Section 4.3.1 remains:

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{5.1c}$$

Finally, since the correct classification of only the basic activities is of interest, the measure accuracy is defined as following:

$$accuracy = \frac{1}{N - \sum_{j=C+1}^{C+B} P_{j,C+1}} \sum_{i=1}^{C} P_{i,i}. \tag{5.1d}$$

The above introduced adjusted performance measures can be applied on both activity recognition tasks of this chapter. It should be noticed that since the 'basic' task does not include any other activities, the adjusted measures reduce to the original measures of Section 4.3.1. Concrete confusion matrices on the defined 'basic' and 'extended' classification problems are shown as results in Section 5.6.1 and Section 5.6.2, respectively. Moreover, those confusion matrices are used to understand the results in more detail and compare different approaches.

5.4.2 Intensity Estimation

Regarding the intensity estimation problem as a 3-class classification task, the performance measures can be defined similarly to that of the activity recognition tasks. The notation of Table 5.2 is similar to the generalized confusion matrix of Table 5.1. Here,

Table 5.3: *General confusion matrix of the intensity estimation task, using annotated activity classes.*

Annotated activity	**Estimated intensity** light	moderate	vigorous
1	$P_{1,1}$	$P_{1,2}$	$P_{1,3}$
2	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$
...			
L	$P_{L,1}$	$P_{L,2}$	$P_{L,3}$
$L+1$	$P_{L+1,1}$	$P_{L+1,2}$	$P_{L+1,3}$
...			
$L+M$	$P_{L+M,1}$	$P_{L+M,2}$	$P_{L+M,3}$
$L+M+1$	$P_{L+M+1,1}$	$P_{L+M+1,2}$	$P_{L+M+1,3}$
...			
$L+M+V$	$P_{L+M+V,1}$	$P_{L+M+V,2}$	$P_{L+M+V,3}$
	R_1	R_2	R_3

S_i refers to the number of samples annotated as either the annotated intensity class light, moderate or vigorous, and R_j refers to the number of samples recognized as one of the intensity classes. Using this notation, the 4 performance measures are defined the following way:

$$precision = \frac{1}{3}\left(\frac{P_{1,1}}{R_1} + \frac{P_{2,2}}{R_2} + \frac{P_{3,3}}{R_3}\right) \tag{5.2a}$$

$$recall = \frac{1}{3}\left(\frac{P_{1,1}}{S_1} + \frac{P_{2,2}}{S_2} + \frac{P_{3,3}}{S_3}\right) \tag{5.2b}$$

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5.2c}$$

$$accuracy = \frac{\sum_{i=1}^{3} P_{i,i}}{\sum_{i=1}^{3}\sum_{j=1}^{3} P_{i,j}}. \tag{5.2d}$$

The drawback of the representation of Table 5.2 is that only the confusion between the 3 intensity classes are shown, no information about *e.g.* the intensity of which specific activities is estimated inaccurately. A more detailed representation can be given when using the annotated activity classes in the confusion matrix, as shown in Table 5.3. For this representation, assume that the 'intensity' task consists of L light effort activities: $1,\ldots,L$ and M moderate effort activities: $1,\ldots,M$ and V vigorous effort activities: $1,\ldots,V$. Moreover, let A be the set of all different activities: $A = L \cup M \cup V$. It is worth to note that the confusion matrix of Table 5.2 can be considered as the result of merging all the rows in Table 5.3 of activities belonging to the same intensity level.

The R_j ($j = 1, \ldots, 3$) number of samples recognized as one of the intensity classes can be determined as

$$R_j = \sum_{i \in A} P_{i,j}, \tag{5.3}$$

while the S_i ($i = 1, \ldots, 3$) number of samples annotated as one of the intensity classes can be determined as follows:

$$S_1 = \sum_{i \in L} (P_{i,1} + P_{i,2} + P_{i,3}) \tag{5.4a}$$

$$S_2 = \sum_{i \in M} (P_{i,1} + P_{i,2} + P_{i,3}) \tag{5.4b}$$

$$S_3 = \sum_{i \in V} (P_{i,1} + P_{i,2} + P_{i,3}). \tag{5.4c}$$

Using the definitions of (5.3) and (5.4), the performance measures precision and recall can be defined as follows:

$$precision = \frac{1}{3}\left(\frac{\sum_{i \in L} P_{i,1}}{R_1} + \frac{\sum_{i \in M} P_{i,2}}{R_2} + \frac{\sum_{i \in V} P_{i,3}}{R_3}\right) \tag{5.5a}$$

$$recall = \frac{1}{3}\left(\frac{\sum_{i \in L} P_{i,1}}{S_1} + \frac{\sum_{i \in M} P_{i,2}}{S_2} + \frac{\sum_{i \in V} P_{i,3}}{S_3}\right). \tag{5.5b}$$

The definition of F-measure remains unaltered:

$$F\text{-}measure = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{5.5c}$$

while the measure accuracy can be determined as

$$accuracy = \frac{\sum_{i \in L} P_{i,1} + \sum_{i \in M} P_{i,2} + \sum_{i \in V} P_{i,3}}{\sum_{i \in A} \sum_{j=1}^{3} P_{i,j}}. \tag{5.5d}$$

Results on the 'intensity' classification task are shown in Section 5.6.3, using the representation form of Table 5.3 and the performance measures as defined by (5.5).

5.5 Evaluation Techniques

This section presents the evaluation methods used to obtain the results in Section 5.6. The goal of the evaluation of the created classifiers is to estimate their behaviour in everyday life scenarios, thus to simulate how they would perform in named situations. The commonly used standard k-fold cross-validation (CV) is not adequate for this task, since it only estimates the behaviour of the scenario in which the classifier was trained, thus on a limited and known set of users and physical activities. Nevertheless, standard 10-fold CV is also applied as an evaluation technique in the experiments of this chapter for comparison reasons. These results will show how "optimistic" k-fold CV is for validation, that is, how unrealistic the so achieved performance is in real life scenarios.

The simulation of everyday life scenarios means concretely to simulate how the created system behaves when used by a previously (in training time) unknown person, or when a previously unknown activity is performed. To simulate subject independency the evaluation technique leave-one-subject-out (LOSO) CV is applied. Since the used PAMAP2 dataset provides data from 9 subjects, LOSO 9-fold CV is applied in the experiments of this chapter. Moreover, to simulate the scenario of performing unknown other activities a new evaluation technique is introduced: leave-one-activity-out (LOAO). The basic idea of LOAO is similar to the LOSO evaluation technique. However, the concrete definition of LOAO for the activity recognition and intensity estimation tasks of this chapter is described in more detail in the next two subsections.

5.5.1 Activity Recognition

In both activity recognition tasks defined in this chapter ('basic' and 'extended') the set of basic activities is known during training time. Therefore, only the simulation of performing unknown other activities within the 'extended' task is required. The concept of the LOAO technique applied on the 'extended' task will thus be referred to as leave-one-other-activity-out (LOOAO) hereafter: for this classification problem including B other activities, data from $B - 1$ other activities is used for training and data from the remaining other activity for testing, repeating this procedure B times leaving always another activity's data for testing.

To receive the best possible understanding of the developed system's behaviour in everyday life scenarios, the newly introduced LOOAO evaluation technique is combined with the LOSO technique. This combined evaluation method will be referred to as LOSO_LOOAO throughout this chapter, the procedure is formally described in Algorithm 5.1. With LOSO_LOOAO evaluation the following practical scenarios are evaluated:

- The system is trained with a large amount of subjects for the 'extended' task. Then the system is deployed to a new subject (thus for this subject no data was available during the training phase of the system), and the new subject performs one of the basic activities (estimated through the LOSO component).

- The system is trained with a large amount of subjects for the 'extended' task. Then the system is deployed to a new subject, who performs one of the known other activities (estimated through the LOSO component). This is the first step in testing the robustness of the system in situations when the user performs activities other than the few basic recognized ones.

- The system is trained with a large amount of subjects for the 'extended' task. Then the system is deployed to a new subject, who performs a previously unknown activity – thus an activity neither belonging to the basic activity classes, nor to one of the other activities available during the training phase (estimated through the LOOAO component). This scenario simulates basically the generalization characteristic of the classifier's other activity model, estimating how

robust the system is in the usually neglected situation when unknown activities are performed.

Algorithm 5.1 LOSO_LOOAO

Require: **S** is the set of S different subjects, $\{s : 1, 2, \ldots, S\}$
C is the set of C different basic activities, $\{c : 1, 2, \ldots, C\}$
B is the set of B different other activities, $\{b : 1, 2, \ldots, B\}$
A is the set of all different activities: $\mathbf{A} = \mathbf{C} \cup \mathbf{B}$, an arbitrary activity is referred to as a
N is the set of N different samples, where each sample consists of subject and activity information and a feature vector, thus $\underline{n}$: $<s, a, features>$
$s(\underline{n})$ refers to the subject of the sample $\underline{n}$
$a(\underline{n})$ refers to the activity of the sample $\underline{n}$

1: **procedure** LOSO_LOOAO(**S,C,B,A,N**)
2: **for** $i \leftarrow 1, S$ **do**
3: $\mathbf{P}_{train} = \{\forall \underline{n} \in \mathbf{N} \,|s(\underline{n}) \neq i\}$
4: $\mathbf{P}_{test} = \{\forall \underline{n} \in \mathbf{N} \,|s(\underline{n}) = i\}$
5: $\mathbf{P}_{test\_basic} = \{\forall \underline{n} \in \mathbf{P}_{test} | a(\underline{n}) \in \mathbf{C}\}$
6: Train classifier using $\mathbf{P}_{train} \rightarrow F_i$
7: Use F_i on $\mathbf{P}_{test\_basic}$ % *LOSO on basic activities*
8: **for** $j \leftarrow 1, B$ **do**
9: $\mathbf{P}_{train\_other} = \{\forall \underline{n} \in \mathbf{P}_{train} | ((a(\underline{n}) \in \mathbf{C}\ ) \vee ((a(\underline{n}) \in \mathbf{B}\ and\ (a(\underline{n}) \neq j)))\}$
% *thus the sample does not belong to the jth other activity*
10: $\mathbf{P}_{test\_other} = \{\forall \underline{n} \in \mathbf{P}_{test} | ((a(\underline{n}) \in \mathbf{B}\ ) \wedge (a(\underline{n}) = j))\}$
% *thus the sample belongs to the jth other activity*
11: Train classifier using $\mathbf{P}_{train\_other} \rightarrow F_{i,j}$
12: Use $F_{i,j}$ on $\mathbf{P}_{test\_other}$ % *LOOAO on jth other activity*
13: **end for** % *LOOAO with all B other activities is finished here*
14: **end for** % *The LOSO results with the basic activities and the LOOAO results with the other activities together return the LOSO_LOOAO result*
15: **end procedure**

5.5.2 Intensity Estimation

In the intensity estimation classification task defined in this chapter all 18 activities are treated equally, there are no basic and other activities distinguished which are handled differently. Therefore, the concept of the LOAO technique is applied on the 'intensity' task to simulate when one of the 18 activities is left out, while all the other activities are known. Moreover, similar to the evaluation of the 'extended' task as presented in the previous subsection, LOAO is combined with the LOSO technique. This combined evaluation method will be referred to as LOSO_LOAO throughout this chapter, the procedure is formally described in Algorithm 5.2. Overall, 3 different evaluation methods are applied for the 'intensity' task: standard 10-fold CV, LOSO 9-fold CV and the newly introduced LOSO_LOAO technique.

Algorithm 5.2 LOSO_LOAO

Require: **S** is the set of S different subjects, $\{s : 1, 2, \ldots, S\}$
A is the set of all 18 different activities, an arbitrary activity is referred to as a
N is the set of N different samples, where each sample consists of subject and activity information and a feature vector, thus $\underline{n}$: $<s, a, features>$
$s(\underline{n})$ refers to the subject of the sample $\underline{n}$
$a(\underline{n})$ refers to the activity of the sample $\underline{n}$

procedure LOSO_LOAO(**S,A,N**)
  for $i \leftarrow 1, S$ **do**
    $\mathbf{P}_{train} = \{\forall \underline{n} \in \mathbf{N} \,|s(\underline{n}) \neq i\}$
    $\mathbf{P}_{test} = \{\forall \underline{n} \in \mathbf{N} \,|s(\underline{n}) = i\}$
    for $j \leftarrow 1, A$ **do**
      $\mathbf{P}_{train\_loao} = \{\forall \underline{n} \in \mathbf{P}_{train} | a(\underline{n}) \neq j\}$
        % thus the sample does not belong to the jth other activity
      $\mathbf{P}_{test\_loao} = \{\forall \underline{n} \in \mathbf{P}_{test} | a(\underline{n}) = j\}$
        % thus the sample belongs to the jth other activity
      Train classifier using $\mathbf{P}_{train\_loao} \rightarrow F_{i,j}$
      Use $F_{i,j}$ on $\mathbf{P}_{test\_loao}$ *% LOAO on jth other activity*
    end for *% LOAO with all activities is finished here for subject i*
  end for *% LOSO with all different subjects is finished here*
end procedure

5.6 Results and Discussion

Table 5.4 shows the confusion matrix on the 'basic' classification task using the C4.5 decision tree classifier and standard CV as evaluation technique (the results are an average of 10 test runs). Almost no misclassifications can be observed, all performance measures are clearly above 99%. Therefore, this result could indicate that physical activity recognition is an easily solvable classification problem, even with a simple classifier such as a decision tree. However, the result of Table 5.4 has two main drawbacks: it is subject dependent (thus does not tell anything about the performance of the system when used by a new subject), and only applies to the specific scenario of these 6 basic activity classes. Therefore, an extension of this result is required to increase the applicability of the system concerning both limitations. Further results in this section will show that the performance of activity monitoring is much lower under realistic, everyday life conditions.

5.6.1 The 'Basic' Classification Task

The 'basic' classification task serves only for comparison, thus to see the baseline characteristic of physical activity recognition. Since all activities of the task are to be recognized, only the subject independency of the system can be simulated from the aforementioned two issues. The performance measures are shown in Table 5.5 for

Table 5.4: *Confusion matrix on the 'basic' task using the C4.5 decision tree classifier and standard CV evaluation technique. The table shows how different annotated activities are classified in [%].*

Annotated activity	**Recognized activity**					
	1	2	3	4	5	6
1 lie	100	0	0	0	0	0
2 sit/stand	0	99.87	0	0	0.13	0
3 walk	0	0.05	99.66	0	0.02	0.27
4 run	0	0	0	100	0	0
5 cycle	0	0.29	0.25	0.11	99.29	0.06
6 Nordic walk	0	0	0.60	0	0	99.40

both standard CV and LOSO evaluation. Each of the tests is performed 10 times, the table shows the mean and standard deviation of these 10 test runs.

The results of Table 5.5 show the significant difference between standard CV and LOSO, for both classifiers. Table 5.6 shows the confusion matrix on the 'basic' task using the C4.5 decision tree and LOSO as evaluation technique. Comparing the confusion matrices of Table 5.4 (F-measure is 99.71%) and Table 5.6 (F-measure is 95.50%) the differences between the results obtained with standard CV and LOSO can be observed in more detail. The recognition rate of all 6 activities decreases with LOSO evaluation, this is the most significant with the activity *Nordic walk*: the performance decreases from 99.40% to 83.19%. The reason for the lower performance can be explained by the diversity in how subjects perform physical activities (*e.g.* the differing pattern and intensity of arm movements during the activities *walk* and *Nordic walk* by different subjects, which leads to the significant confusion between these two activities in Table 5.6). Subject independent evaluation simulates this behaviour, while subject dependent evaluation ignores it. Therefore, the latter method leads to highly "optimistic" results as observed in Table 5.5, and will be shown in Table 5.7 and Table 5.12 on the classification tasks 'extended' and 'intensity', respectively.

An interesting result in Table 5.5 is that the AdaBoost.M1 classifier only slightly outperforms the C4.5 classifier on the 'basic' task (the difference between the two classifiers on the 'extended' task is much more significant, as shown in the next subsection). This can be explained by the fact that the 'basic' task is a rather simple classification problem where even base-level classifiers can reach the highest possible accuracy. Therefore, it is not necessarily worth using more complex classification algorithms here. The lower performance when using LOSO evaluation is due to the difficulty of the generalization in respect of the users, and not due to the difficulty of the classification task.

Although using subject independent evaluation is the first step towards simulating the conditions of everyday usage of activity monitoring applications, the 'basic' task only estimates the system's behaviour when activities of one of the 6 included activity classes are performed, thus the system's response is not defined when the user performs activities such as *descend stairs* or *vacuum clean*. This issue is discussed in the next subsection, by analyzing the results obtained on the 'extended' task.

Table 5.5: *Performance measures on the 'basic' activity recognition task. The results are averaged over* 10 *test runs, mean and standard deviation is given for each experimental setup.*

Classifier	Evaluation method	Precision	Recall	F-measure	Accuracy
C4.5	standard CV	99.71 ± 0.04	99.70 ± 0.02	99.71 ± 0.03	99.71 ± 0.03
	LOSO	96.05 ± 1.06	94.96 ± 1.40	95.50 ± 1.20	95.14 ± 1.10
AdaBoost.M1	standard CV	99.97 ± 0.02	99.97 ± 0.02	99.97 ± 0.02	99.97 ± 0.02
	LOSO	95.91 ± 1.45	95.47 ± 1.45	95.69 ± 1.40	95.43 ± 1.54

Table 5.6: *Confusion matrix on the 'basic' task using the C4.5 decision tree classifier and LOSO CV evaluation technique. The table shows how different annotated activities are classified in [%].*

Annotated activity	**Recognized activity**					
	1	2	3	4	5	6
1 lie	97.56	2.33	0	0	0.12	0
2 sit/stand	0.04	98.24	0.80	0	0.93	0
3 walk	0	2.58	94.97	0	0.75	1.71
4 run	0	0	0	98.24	1.69	0.07
5 cycle	0	1.94	0.29	0.14	97.56	0.07
6 Nordic walk	0	0	16.79	0.02	0	83.19

5.6.2 The 'Extended' Classification Task

The performance measures on the 'extended' task are presented in Table 5.7: for each of the 4 other activity models, by using the 2 classifiers and the 3 different evaluation techniques. The results are given in form of mean and standard deviation of the 10 test runs performed for every possible combination of the models, classifiers and evaluation methods. Overall it is clear that with the inclusion of the other activities the classification task becomes significantly more difficult (*cf.* the comparison of the results achieved with standard CV and LOSO to the respective results on the 'basic' task in Table 5.5). This can be explained not only by the increased number of activities in the classification problem (it should be noted that the defined performance measures for the 'extended' task only focus on the basic activity classes, thus the results are comparable with that of the 'basic' task), but also by the fact that the characteristic of some of the introduced other activities overlap with the characteristic of some of the basic activity classes. For example, the other activity *iron* has a similar characteristic to talking and gesticulating during *stand*, thus misclassifications appear between these two activities. Similarly it is nontrivial to distinguish running with a ball (during the other activity *play soccer*) from just *running*. Since the 'extended' task defines a complex classification problem, it is worth to apply more complex classification algorithms here – contrary to the 'basic' classification task. For example when considering the 'allSeparate' model and LOSO evaluation, the C4.5 decision tree only achieves an F-measure of 83.30% while with the AdaBoost.M1 classifier 92.22% can be reached.

From the results of Table 5.7 it is obvious that the performance measures achieved with LOSO evaluation are significantly lower than results obtained with standard CV, as already seen in Table 5.5 and explained in Section 5.6.1. If only considering subject independency the 'allSeparate' model performs best, closely followed by the models 'preReject' and 'bgClass'. However, on the 'extended' task it is also simulated when the user of the system performs unknown other activities (LOOAO). The results of applying the evaluation method of Algorithm 5.1 are shown in Table 5.7 in the respective rows of LOSO_LOOAO. Considering this combined evaluation technique the 'bgClass' model performs best, followed by the models 'preReject' and 'allSeparate'. From all the 4 other activity models the 'allSeparate' model shows the largest decrease

Table 5.7: *Performance measures on the 'extended' activity recognition task. The results are averaged over* 10 *test runs, mean and standard deviation is given for each experimental setup.*

Model	Classifier	Evaluation method	Precision	Recall	F-measure	Accuracy
'allSeparate'	C4.5	standard CV	98.17 ± 0.23	98.00 ± 0.09	98.09 ± 0.14	95.80 ± 0.25
		LOSO	89.77 ± 1.89	77.75 ± 3.08	83.30 ± 2.10	73.81 ± 2.21
		LOSO_LOOAO	81.84 ± 1.77	78.59 ± 3.43	80.16 ± 2.44	67.06 ± 2.71
	AdaBoost.M1	standard CV	99.94 ± 0.01	99.93 ± 0.04	99.93 ± 0.02	99.83 ± 0.05
		LOSO	95.42 ± 0.98	89.23 ± 2.00	92.22 ± 1.40	86.60 ± 2.09
		LOSO_LOOAO	86.80 ± 0.99	88.72 ± 1.28	87.75 ± 1.07	78.83 ± 1.29
'bgClass'	C4.5	standard CV	98.68 ± 0.17	98.66 ± 0.11	98.67 ± 0.12	96.85 ± 0.21
		LOSO	89.85 ± 1.35	85.83 ± 3.11	87.78 ± 2.11	80.63 ± 1.81
		LOSO_LOOAO	83.64 ± 2.46	85.56 ± 2.67	84.58 ± 2.39	73.76 ± 2.10
	AdaBoost.M1	standard CV	99.96 ± 0.02	99.88 ± 0.03	99.92 ± 0.02	99.77 ± 0.05
		LOSO	96.07 ± 0.99	85.76 ± 2.45	90.61 ± 1.72	84.14 ± 2.35
		LOSO_LOOAO	91.81 ± 0.82	86.82 ± 1.71	89.24 ± 1.17	80.97 ± 1.20
'preReject'	C4.5	standard CV	98.28 ± 0.14	97.83 ± 0.12	98.05 ± 0.07	95.46 ± 0.14
		LOSO	88.58 ± 1.40	78.66 ± 2.51	83.30 ± 1.36	71.78 ± 1.76
		LOSO_LOOAO	83.07 ± 1.68	78.83 ± 3.63	80.87 ± 2.53	67.32 ± 2.74
	AdaBoost.M1	standard CV	99.95 ± 0.04	99.89 ± 0.04	99.92 ± 0.04	99.82 ± 0.06
		LOSO	93.85 ± 1.57	88.46 ± 2.26	91.07 ± 1.83	85.20 ± 2.07
		LOSO_LOOAO	87.99 ± 1.47	87.98 ± 1.80	87.98 ± 1.58	79.11 ± 1.60
'postReject'	C4.5	standard CV	99.08 ± 0.09	98.21 ± 0.15	98.64 ± 0.10	96.89 ± 0.20
		LOSO	92.93 ± 0.93	77.65 ± 3.05	84.59 ± 2.11	74.89 ± 1.80
		LOSO_LOOAO	89.02 ± 0.62	78.96 ± 2.05	83.67 ± 1.23	71.59 ± 1.66
	AdaBoost.M1	standard CV	99.93 ± 0.04	99.82 ± 0.02	99.87 ± 0.03	99.75 ± 0.05
		LOSO	95.76 ± 1.38	81.18 ± 2.57	87.86 ± 1.87	80.92 ± 2.50
		LOSO_LOOAO	92.01 ± 1.80	80.65 ± 3.02	85.94 ± 2.40	77.78 ± 2.52

in performance from LOSO evaluation to LOSO_LOOAO evaluation. Especially the precision measure decreases largely, thus when the user performs unknown activities they are more likely recognized as one of the basic activity classes compared to the results of other models. This behaviour can be observed when comparing Table 5.8 (LOSO evaluation) with Table 5.9 (LOSO_LOOAO evaluation). The recognition rate of the basic activities is similar in both cases since their evaluation method is the same. However, the recognition rate of all other activities decreases significantly in Table 5.9, *e.g. descend stairs* from 99.21% to 70.81% or *rope jump* from 99.24% to 91.90%. This significant decrease in the precision measure can be explained by the fact that for the 'allSeparate' model separate activity classes are created and trained for each of the known other activities, thus the generalization capability of the model is rather limited when a previously unknown activity is performed. On the other hand, the training instances belonging to the other/background activity class of the 'bgClass' model are scattered in the feature space, resulting in a large class with good generalization characteristic. Moreover, since much more instances are used for the creation of the background activity class during training than for the 6 basic activity classes, this class becomes more important, thus resulting in significantly higher precision than recall result with the 'bgClass' model.

The 'preReject' other activity model performed second best in both the LOSO and the LOSO_LOOAO evaluation, justifying the idea of first recognizing whether a performed activity belongs to the basic activity classes or not. When analyzing the trained classifiers for the two levels of this model, it can be noticed that the classifier of the first level is much more complex: although representing only a binary decision, the separation of basic activities from other activities is a difficult task. The classification problem defined in the second level of the model is identical to the 'basic' classification task defined in this chapter, and thus is – as discussed in the previous subsection – a rather simple task. Finally, the 'postReject' model performed worst with both LOSO and LOSO_LOOAO evaluation, resulting in the lowest F-measure and accuracy values. Since the basic activities are distinguished on the first level of this model (without any other activities concerned), this model has the least confusion between the basic activity classes. The confusion matrices belonging to the evaluation of this model – one example is given in Table 5.10 – confirm this statement: except for some misclassifications of *Nordic walk* samples as *normal walk*, all confusion is done towards the other activity class. Moreover, due to the unbalanced classification tasks defined on the second level of the model (only one basic activity versus all other activities, thus these tasks are even more unbalanced than the classification task defined by the 'bgClass' model), the precision values are comparable with those of other models. Therefore, if the goal of an activity recognition application is only the precise recognition of activities of interest the 'postReject' model can also be considered, but otherwise one of the three other models should be used.

From the results of Table 5.7 the performance measures obtained with LOSO_LOOAO evaluation should be regarded as most important, since this evaluation technique simulates the widest range of practical scenarios. The approach achieving the best performance results with LOSO_LOOAO can thus be regarded as the approach which is the most robust in everyday life situations. Therefore, overall the

Table 5.8: *Confusion matrix on the 'extended' task using the 'allSeparate' model, AdaBoost.M1 classifier and LOSO evaluation technique. The table shows how different annotated activities are classified in [%].*

Annotated activity	**Recognized activity**						
	1	2	3	4	5	6	0
1 lie	97.35	0.71	0	0	0	0	1.94
2 sit/stand	0.03	91.93	0	0	0	0	8.04
3 walk	0	0	89.08	0	0	0.21	10.71
4 run	0	0	0	73.76	0	0.01	26.23
5 cycle	0	0.01	0.03	0	96.05	0.06	3.85
6 Nordic walk	0	0	6.17	0	0.02	87.21	6.59
7 drive car	0	47.64	0	0	0.83	0	51.54
8 asc. stairs	0	0	0.62	0	0	0	99.38
9 desc. stairs	0	0	0	0	0.79	0	99.21
10 vacuum clean	0	0.01	0	0	0.16	0	99.83
11 iron	0	2.83	0	0	0.03	0	97.14
12 fold laundry	0	5.33	0	0	0.02	0	94.65
13 clean house	0.02	5.53	0.01	0	0.31	0	94.13
14 play soccer	0	0	2.48	10.66	0	0.28	86.58
15 rope jump	0	0	0	0.76	0	0	99.24

Table 5.9: *Confusion matrix on the 'extended' task using the 'allSeparate' model, AdaBoost.M1 classifier and LOSO_LOOAO evaluation technique. The table shows how different annotated activities are classified in [%].*

Annotated activity	**Recognized activity**						
	1	2	3	4	5	6	0
1 lie	97.37	1.02	0	0	0	0	1.60
2 sit/stand	0.12	90.95	0	0	0	0	8.93
3 walk	0	0	91.22	0	0	0.22	8.55
4 run	0	0	0	75.89	0	0	24.11
5 cycle	0	0	0.06	0	95.11	0.19	4.64
6 Nordic walk	0	0	12.54	0	0	81.78	5.68
7 drive car	0.02	49.67	0	0	0.12	0	50.19
8 asc. stairs	0	0	12.49	0	0.91	1.48	85.12
9 desc. stairs	0	0	9.29	0.02	17.92	1.97	70.81
10 vacuum clean	0	0.02	0	0	9.17	0	90.81
11 iron	0	22.02	0	0	0.00	0	77.97
12 fold laundry	0	6.53	0	0	0.03	0	93.44
13 clean house	0.09	9.71	0.02	0	0.35	0	89.84
14 play soccer	0	0	4.66	31.67	0	2.43	61.24
15 rope jump	0	0	0.35	6.33	1.41	0	91.90

Table 5.10: *Confusion matrix on the 'extended' task using the 'postReject' model, AdaBoost.M1 classifier and LOSO_LOOAO evaluation technique. The table shows how different annotated activities are classified in [%].*

Annotated activity	**Recognized activity**						
	1	2	3	4	5	6	0
1 lie	88.58	0.45	0	0	0	0	10.98
2 sit/stand	0.31	89.79	0	0	0	0	9.90
3 walk	0	0	81.29	0	0	0.13	18.58
4 run	0	0	0	56.61	0	0.30	43.09
5 cycle	0	0	0	0	91.82	0.03	8.15
6 Nordic walk	0	0	6.00	0	0	75.81	18.19
7 drive car	0	36.70	0	0	0	0	63.30
8 asc. stairs	0	0	0.78	0	0	0	99.22
9 desc. stairs	0	0	0.22	0	1.22	0.33	98.23
10 vacuum clean	0	0	0	0	0.69	0	99.31
11 iron	0	16.54	0	0	0.05	0	83.41
12 fold laundry	0	2.35	0	0	0	0	97.65
13 clean house	0.41	6.37	0	0	0.09	0	93.13
14 play soccer	0	0	3.27	25.22	0	1.77	69.75
15 rope jump	0	0	0.03	2.91	0	0	97.07

'bgClass' model can be regarded as the model with the best generalization characteristic: the approach using the 'bgClass' model and the AdaBoost.M1 classifier achieves an average F-measure of 89.24% and an average accuracy of 80.97%. The confusion matrix obtained with this approach is shown in Table 5.11 (the results represent the average from the 10 test runs). It is obvious that most of the misclassifications occur due to the other activities: either a sample belonging to a basic activity class is classified into the background class, or a sample from an other activity is confused with one of the basic activities. For example, *drive car* and *iron* are in high percentage confused with the basic class *sit/stand*. This is due to the overlapping characteristic of some basic and other activities, as already discussed above. The strength of the 'bgClass' model is especially pointed out by the results obtained with other activities such as *ascend stairs*, *descend stairs*, *vacuum clean* or *rope jump*: although previously unknown to the system, these activities were basically not misclassified as a basic activity. Therefore, it can be expected that the proposed approach shows such robustness with most of other unknown activities as well. Only unknown activities similar to the target activities might be problematic for the 'bgClass' approach, as seen with *drive car* or *iron*, or is expected with activities such as *computer work* or *watch TV*. However, it is difficult to set the defining boundaries of some of the basic activity classes – *e.g.* if *computer work* should be regarded as *sitting* or as a separate other class. Deciding this question might highly depend on the actual application.

Table 5.11: *Confusion matrix on the 'extended' task using the 'bgClass' model, AdaBoost.M1 classifier and LOSO_LOOAO evaluation technique. The table shows how different annotated activities are classified in [%].*

Annotated activity	**Recognized activity**						
	1	2	3	4	5	6	0
1 lie	96.66	2.62	0	0	0	0	0.72
2 sit/stand	0.15	90.05	0	0	0	0	9.80
3 walk	0	0	85.87	0	0	0.13	14.00
4 run	0	0	0.16	76.24	0	0.30	23.31
5 cycle	0	0	0.01	0	92.43	0.03	7.53
6 Nordic walk	0	0	8.71	0	0	79.69	11.60
7 drive car	0	39.10	0	0	0.06	0	60.84
8 asc. stairs	0	0	0.53	0	0	0.01	99.46
9 desc. stairs	0.08	0	1.97	0.02	0.84	0.06	97.04
10 vacuum clean	0	0	0	0	0.42	0	99.58
11 iron	0	20.02	0	0	0.01	0	79.97
12 fold laundry	0	3.70	0.01	0	0	0	96.29
13 clean house	0.26	7.10	0	0	0.06	0	92.58
14 play soccer	0	0	3.34	32.08	0	0.13	64.46
15 rope jump	0	0	0.11	0.11	0	0	99.78

5.6.3 The 'Intensity' Classification Task

The performance measures on the 'intensity' task are presented in Table 5.12: for each of the 3 evaluation techniques, using the 2 classifiers. The results are given in form of mean and standard deviation of the 10 test runs performed for every possible combination of the classifiers and evaluation methods. Similar to the classification tasks 'basic' and 'extended', the performance measures achieved with LOSO evaluation are significantly lower than results obtained with standard CV. Moreover, with the simulation of performing previously unknown activities there is a further large decrease in performance. This can be observed in detail when comparing the confusion matrices of Table 5.13 (LOSO evaluation) and Table 5.14 (LOSO_LOAO evaluation), both achieved with the AdaBoost.M1 classifier. When only considering subject independency, the intensity of most activities is estimated reliably (*cf.* Table 5.13). However, with the introduction of the LOAO component the intensity estimation of some activities completely fail, *e.g.* only 2.20% of the *ascend stairs* samples or only 12.19% of the *vacuum clean* samples are estimated correctly. On the other hand, the intensity of the different posture-related activities – such as *sit*, *computer work* or *watch TV* – is estimated well. This can be explained by the fact that, although a certain activity itself is unknown, during the training of the classifier enough samples from similar activities are available.

Table 5.12: *Performance measures on the 'intensity' classification task. The results are averaged over* 10 *test runs, mean and standard deviation is given for each experimental setup.*

Classifier	Evaluation method	Precision	Recall	F-measure	Accuracy
C4.5	standard CV	97.50 ± 0.19	97.35 ± 0.22	97.42 ± 0.19	97.77 ± 0.11
	LOSO	88.86 ± 1.08	89.93 ± 0.43	89.39 ± 0.66	91.70 ± 0.41
	LOSO_LOAO	56.92 ± 3.02	58.29 ± 3.21	57.59 ± 3.09	70.07 ± 1.87
AdaBoost.M1	standard CV	99.88 ± 0.03	99.84 ± 0.04	99.86 ± 0.04	99.85 ± 0.04
	LOSO	93.76 ± 1.33	94.99 ± 0.62	94.37 ± 0.96	95.04 ± 0.53
	LOSO_LOAO	62.87 ± 0.94	65.70 ± 0.80	64.25 ± 0.86	73.93 ± 0.63

Table 5.13: *Confusion matrix on the 'intensity' classification task using the AdaBoost.M1 classifier and LOSO evaluation technique. The table shows how different annotated activities are classified in [%].*

Annotated activity	**Estimated intensity**		
	light	moderate	vigorous
1 lie	99.97	0.03	0
2 sit	99.98	0.02	0
3 stand	99.98	0.02	0
4 walk	0.02	96.38	3.60
5 run	0	0.07	99.93
6 cycle	0.44	99.45	0.11
7 Nordic walk	0	96.42	3.58
8 watch TV	100.00	0	0
9 computer work	100.00	0	0
10 drive car	97.26	2.74	0
11 asc. stairs	0.62	7.89	91.49
12 desc. stairs	1.01	93.23	5.75
13 vacuum clean	19.61	80.06	0.33
14 iron	98.85	1.15	0
15 fold laundry	90.36	9.64	0
16 clean house	72.97	26.74	0.29
17 play soccer	0	8.15	91.85
18 rope jump	0	0.03	99.97

Overall, the results on the 'intensity' task presented in this subsection (especially the results in Table 5.14) show that the intensity estimation when subjects perform previously unknown activities can be highly unreliable. On the one hand, this emphasizes on the importance of applying LOSO_LOAO evaluation, thus that the simulation of a trained classifier's performance on unknown activities should not be neglected. On the other hand, these results also encourage to develop more robust approaches for the intensity estimation of physical activities, such that they have better generalization characteristics.

5.7 Conclusion

This chapter developed the means for simulating everyday life scenarios and thus to evaluate the robustness of activity recognition and intensity estimation – a usually neglected point of view in the development of physical activity monitoring systems. Experiments were carried out on classification problems defined on the recently released PAMAP2 physical activity monitoring dataset. An activity recognition task was defined, including 6 basic activity classes and 9 different other activities. The goal of this classification task was the accurate recognition and separation of the ba-

Table 5.14: *Confusion matrix on the 'intensity' classification task using the AdaBoost.M1 classifier and LOSO_LOAO evaluation technique. The table shows how different annotated activities are classified in [%].*

Annotated activity	**Estimated intensity**		
	light	moderate	vigorous
1 lie	99.97	0.03	0
2 sit	99.99	0.01	0
3 stand	99.86	0.14	0
4 walk	0.26	44.81	54.93
5 run	0	3.28	96.72
6 cycle	52.52	47.07	0.41
7 Nordic walk	0	74.59	25.41
8 watch TV	100.00	0	0
9 computer work	100.00	0	0
10 drive car	97.72	2.28	0
11 asc. stairs	0.73	97.07	2.20
12 desc. stairs	2.98	25.18	71.84
13 vacuum clean	87.27	12.19	0.54
14 iron	96.14	3.86	0
15 fold laundry	90.91	9.09	0
16 clean house	45.96	53.50	0.55
17 play soccer	0.18	29.49	70.33
18 rope jump	0	3.21	96.79

sic activities, while samples of the other activities should be recognized as part of an other activity class or should be rejected. Moreover, an intensity estimation task was defined including all 18 activities from the PAMAP2 dataset. The goal of this classification task was to distinguish activities of light, moderate and vigorous effort. Common data processing and classification methods were used to achieve the classification goals, comparing two – in previous work successfully applied – classification algorithms: the C4.5 decision tree classifier and the AdaBoost.M1 algorithm. Moreover, to deal with other activities in the activity recognition task, 4 different models are proposed: 'allSeparate', 'bgClass', 'preReject' and 'postReject'. Finally, the evaluation of the proposed methods was performed with different techniques, including standard CV, LOSO and the newly introduced LOAO. Standard 10-fold CV was only included for comparison reasons: to underline how unrealistic the so achieved performance is in everyday life scenarios. The LOSO technique serves to simulate subject independency, while LOAO simulates the scenario of performing unknown other activities. Considering the activity recognition task, the results of the thorough evaluation process revealed that the 'bgClass' model has the best generalization characteristic, while the generalization capability of the widely used 'allSeparate' approach is rather limited in respect of recognizing previously unknown activities. As for the intensity

estimation task, the results showed that classification can be highly unreliable when dealing with previously unknown activities, thus encouraging to improve existing approaches.

Developing physical activity monitoring systems while also taking *e.g.* subject independency or unknown activities into account has two important benefits compared to when standard CV evaluation is used only. First of all it estimates how the developed system behaves in various everyday life scenarios, while this behaviour would be otherwise undefined. Moreover, the best performing models and algorithms can be selected when applying LOSO and LOAO evaluation during the development phase of the system, hence creating the best possible system from the robustness point of view for everyday life. In future work it is planned to investigate how well the developed approaches generalize with user groups (*e.g.* elderly) significantly differing from the subjects (all young, healthy adults) included in the PAMAP2 dataset. Moreover, it is also planned to investigate the effect of increasing the number of known (thus in the training included) other activities, with the goal to increase the robustness towards unknown other activities even more while keeping the high performance regarding the basic activity classes.

6

Confidence-based Multiclass AdaBoost

6.1 Introduction

The use of meta-level classifiers for physical activity monitoring problems is not as widespread as using different base-level classifiers. However, comparing base-level and meta-level classifiers on different activity recognition tasks shows that meta-level classifiers (such as boosting, bagging, plurality voting, etc.) outperform base-level classifiers [131]. A complex activity recognition problem including 13 different physical activities is used to evaluate the most widely used base-level (decision trees, k-Nearest Neighbors (kNN), Support Vector Machines (SVM) and Naive Bayes classifiers) and meta-level (bagging, boosting) classifiers [137]. Best performance was achieved with a boosted decision tree classifier. The benchmark results on the PAMAP2 dataset in Section 4.4 confirm that using a boosted C4.5 decision tree classifier is one of the most promising methods for physical activity monitoring.

The boosted decision tree classifier has – apart from good performance results as mentioned above – further benefits: it is a fast classification algorithm with a simple structure, and is therefore easy to implement. These benefits are especially important for physical activity recognition applications since they are usually running on mobile, portable systems for everyday usage, thus the available computational power is limited.[1] Section 8.3.2 will show the feasibility of using boosted decision tree classifier for physical activity monitoring on a mobile platform. Moreover, boosting decision trees has been widely and successfully used in other research fields, *e.g.* recently in multi-task learning [45]. Therefore, considering all the above mentioned benefits, this chapter focuses on using boosting, and in particular using boosted decision tree classifiers for physical activity monitoring.

The benchmark results on the PAMAP2 dataset reveal that the difficulty of the more complex tasks exceeds the potential of existing classifiers. Moreover, the re-

[1] This is the reason why *e.g.* kNN (which also showed generally good performance results on activity recognition tasks) is not further considered here: it is a computationally intensive algorithm, even the more advanced versions of it where the number of distance comparisons is reduced.

sults in Chapter 5 show rather low performance when fully simulating how the most common classifiers perform in everyday life scenarios: None of the results reached an F-measure of 90% on the 'extended' task when using LOSO_LOOAO evaluation technique. Therefore, there is a reasonable demand for modifying and improving existing algorithms. This chapter proposes a confidence-based extension of the well-known AdaBoost.M1 algorithm, called ConfAdaBoost.M1. It builds on established ideas of existing boosting methods. The main contribution of this chapter is thus the ConfAdaBoost.M1 algorithm itself and to show that ConfAdaBoost.M1 significantly improves the results of previous boosting algorithms.

This chapter is organized in the following way: Section 6.2 gives an overview of existing boosting algorithms, highlighting their benefits and drawbacks. The new ConfAdaBoost.M1 algorithm is introduced in Section 6.3. In Section 6.4 the new algorithm is evaluated on various benchmark datasets from the UCI machine learning repository, comparing it to the most commonly used existing boosting methods. Section 6.5 presents the evaluation on a complex activity recognition and intensity estimation problem defined on the PAMAP2 dataset. The main motivation for presenting the ConfAdaBoost.M1 algorithm is the better performance it achieves, compared to existing algorithms, on activity monitoring classification tasks. Finally, the chapter is summarized in Section 6.6.

6.2 Boosting Methods: Related Work

Boosting is a widely used and very successful technique for solving classification problems.[2] The idea behind boosting is to iteratively learn weak classifiers by manipulating the training dataset, and then combine the weak classifiers into a final strong classifier. Contrary to another ensemble learning method, bagging [25] – where the training dataset is sampled with replacement to produce the training instances for each iteration – boosting uses all instances at each repetition. It introduces a weight for each instance in the training dataset, which reflects the instance's importance. The training dataset is reweighted after each iteration, adjusting the weights so that the weak learners focus on the previously misclassified, difficult instances. The final strong classifier is constructed from the weighted combination of weak learners, defining the weights of the single learning models on the basis of their accuracy.

6.2.1 Binary Classification

Boosting was introduced in the computational learning theory literature in the early and mid 90's [54, 55, 152]. To improve a single classifier (weak learner), the first versions of boosting trained additional similar classifiers on filtered versions of the training dataset and produced a majority vote, thus "boosting" the performance [54, 152]. The adaptive boosting algorithm – called AdaBoost – evolved from these algorithms [55], and became the most commonly used technique of boosting, from which many versions have been developed. Moreover, AdaBoost is considered as one

[2] For an extension of boosting to regression problems, the reader is referred to *e.g.* the AdaBoost.R algorithm [55] or to the work by Avnimelech and Intrator [9].

Algorithm 6.1 Discrete AdaBoost

Require: Training dataset of N instances: $(\underline{x}_i, y_i)$ $i = 1, \ldots, N$ ($\underline{x}_i$: feature vector, $y_i \in \{-1, +1\}$)
New instance to classify: $\underline{x}_n$

procedure TRAINING($(\underline{x}_i, y_i)$ $i = 1, \ldots, N$)
  Assign equal weight to each training instance: $w_i = \frac{1}{N}$, $i = 1, \ldots, N$
  for $t \leftarrow 1, T$ **do**
    Fit weak learner on the weighted dataset: $f_t(\underline{x}) \in \{-1, +1\}$
    Compute error e_t of weak learner on weighted dataset: $e_t = \sum_{i:y_i \neq f_t(\underline{x}_i)} w_i$
    Compute $\alpha_t = \log \frac{1-e_t}{e_t}$
    for $i \leftarrow 1, N$ **do**
      if $y_i \neq f_t(\underline{x}_i)$ **then**
        $w_i \leftarrow w_i e^{\alpha_t}$
      end if
    end for
    Normalize the weight of all instances so that $\sum_i w_i = 1$
  end for
end procedure

procedure PREDICTION($\underline{x}_n$)
  The output class is: $\text{sign}[\sum_{t=1}^{T} \alpha_t f_t(\underline{x}_n)]$
end procedure

of the most important ensemble methods, and is named one of the top 10 data mining algorithms by Wu et al. [194].

Already the first version of AdaBoost defines the main ideas of the boosting technique [55]. Assume that a training dataset of N instances is given: $(\underline{x}_i, y_i)$ $i = 1, \ldots, N$ ($\underline{x}_i$ is the feature vector, $y_i \in \{-1, +1\}$). The algorithm trains the weak learners $f_t(\underline{x})$ on weighted versions of the training dataset, giving higher weight to instances that are currently misclassified. This is done for a predefined T number of iterations. The final classifier is a linear combination of the weak learners from each iteration, weighted according to their error rate on the training dataset. This first version of the AdaBoost algorithm was only designed for binary classification problems. As a weak learner, any kind of classifier can be used as long as it is better than random guessing. However, this version of AdaBoost only uses the binary output of the weak learners (-1 or $+1$), thus was called Discrete AdaBoost in [58]. The algorithm is shown in Algorithm 6.1.

A generalization of Discrete AdaBoost is to use real-valued predictions of the weak learners rather than the $\{-1, +1\}$ output. Friedman et al. [58] introduced the algorithm Real AdaBoost, shown in Algorithm 6.2. In this version of AdaBoost the weak learners return a class probability estimate $p_t(\underline{x})$ in each boosting iteration, from which the classification rule $f_t(\underline{x})$ is derived. The sign of $f_t(\underline{x})$ gives the classification prediction, and $|f_t(\underline{x})|$ gives a measure of how confident the weak learner is in the prediction. Experiments by Friedman et al. [58] on various datasets from the UCI machine learning

Algorithm 6.2 Real AdaBoost

Require: Training dataset of N instances: $(\underline{x}_i, y_i)$ $i = 1, \ldots, N$ ($\underline{x}_i$: feature vector, $y_i \in \{-1, +1\}$)
New instance to classify: $\underline{x}_n$

procedure TRAINING($(\underline{x}_i, y_i)$ $i = 1, \ldots, N$)
  Assign equal weight to each training instance: $w_i = \frac{1}{N}$, $i = 1, \ldots, N$
  for $t \leftarrow 1, T$ **do**
    Fit weak learner on the weighted dataset to obtain a class probability estimate: $p_t(\underline{x}) = \hat{P}_w(y = 1|\underline{x}) \in [0, 1]$
    Compute $f_t(\underline{x}) = \frac{1}{2} \log \frac{p_t(\underline{x})}{1-p_t(\underline{x})}$
    for $i \leftarrow 1, N$ **do**
      $w_i \leftarrow w_i e^{-y_i f_t(\underline{x}_i)}$
    end for
    Normalize the weight of all instances so that $\sum_i w_i = 1$
  end for
end procedure

procedure PREDICTION($\underline{x}_n$)
  The output class is: $\text{sign}[\sum_{t=1}^{T} f_t(\underline{x}_n)]$
end procedure

repository, [12], show that this confidence-based version of AdaBoost outperforms the original Discrete AdaBoost algorithm. However, Real AdaBoost is limited to binary classification problems as well.

Apart from Discrete and Real AdaBoost, further boosting methods have been developed for the binary classification case the past decade. Friedman et al. [58] show that the Discrete and Real AdaBoost algorithms can be interpreted as stage-wise estimation procedures for fitting an additive logistic regression model, optimizing an exponential criterion which to second order is equivalent to the binomial log-likelihood criterion. Based on this interpretation of AdaBoost, they introduce the LogitBoost algorithm, which optimizes a more standard (the Bernoulli) log-likelihood. Moreover, Friedman et al. [58] also present the Gentle AdaBoost algorithm, a modified version of Real AdaBoost. It uses Newton stepping rather than exact optimization at each boosting iteration. Another variant of Real AdaBoost – that uses a weighted emphasis function – is presented in [60], called Emphasis Boost. Finally, the Modest AdaBoost algorithm is mentioned here [184]. It not only considers the updated weight distribution for training a classification rule in each boosting step, but also considers the inverse weight distribution to decrease a weak learner's contribution if it works "too good" on data that has already been correctly classified with high margin. As a result, although the training error decreases slower than for comparable methods, Modest AdaBoost produces less generalization error.

Algorithm 6.3 Real AdaBoost.MH

Require: Training dataset of N instances: $(\underline{x}_i, y_i)$ $i = 1, \ldots, N$ ($\underline{x}_i$: feature vector, $y_i \in [1, \ldots, C]$)
New instance to classify: $\underline{x}_n$

procedure TRAINING($(\underline{x}_i, y_i)$ $i = 1, \ldots, N$)
  Transform the C class problem into a binary classification problem, the NC new instances are: $([\underline{x}_i, 1], y_{i1}), ([\underline{x}_i, 2], y_{i2}), \ldots, ([\underline{x}_i, C], y_{iC})$ $i = 1, \ldots, N$ and $y_{ic} \in \{-1, +1\}$ according to the original class response of $\underline{x}_i$
  Assign equal weight to each training instance: $w_k = \frac{1}{NC}$, $k = 1, \ldots, NC$
  for $t \leftarrow 1, T$ **do**
    Fit weak learner on the weighted dataset to obtain a class probability estimate: $p_t(\underline{x}, c) = \hat{P}_w(y = 1 | (\underline{x}, c)) \in [0, 1]$
    Compute $f_t(\underline{x}, c) = \frac{1}{2} \log \frac{p_t(\underline{x}, c)}{1 - p_t(\underline{x}, c)}$
    for $k \leftarrow 1, NC$ **do**
      $w_k \leftarrow w_k e^{-y_{ic} f_t(\underline{x}_i, c)}$
    end for
    Normalize the weight of all instances so that $\sum_k w_k = 1$
  end for
end procedure

procedure PREDICTION($\underline{x}_n$)
  Create C instances out of $\underline{x}_n$: $(\underline{x}_n, 1), \ldots, (\underline{x}_n, C)$
  The output class is: $\arg\max_c \sum_{t=1}^{T} f_t(\underline{x}_n, c) \quad c = 1, \ldots, C$
end procedure

6.2.2 Pseudo-multiclass Classification

The first extensions of AdaBoost for multiclass classification problems can be regarded as pseudo-multiclass solutions: they reduce the multiclass problem into multiple two class problems [153, 155]. One of the most common solutions using binary boosting methods for multiclass problems is AdaBoost.MH, introduced by Schapire and Singer [155]. It converts a C class problem into that of estimating a two class classifier on a training set C times as large, by adding a new "feature" which is defined by the class labels. Thus the original number of N instances is expanded into NC instances. On this new, augmented dataset a binary AdaBoost method (*e.g.* Discrete or Real AdaBoost) can then be applied. Algorithm 6.3 shows the Real AdaBoost.MH algorithm: the extension of the previously presented Real AdaBoost algorithm for the multiclass case using the AdaBoost.MH technique.

There exist other solutions to reduce the multiclass problem into multiple binary classification problems. Schapire [153] combined error-correcting output codes (ECOC) with the original binary AdaBoost method to solve multiclass problems, resulting in the AdaBoost.MO algorithm. Friedman et al. [58] showed how the binary LogitBoost algorithm can be applied for the multiclass case by introducing a "class feature" similar to the AdaBoost.MH method. In [153] experimental results are given

comparing a few pseudo-multiclass algorithms on a set of benchmark problems from the UCI repository, which show that Real AdaBoost.MH performs best amongst these methods.

However, reducing the multiclass classification problem into multiple two class problems has several drawbacks. For instance, as the class label becomes a regular feature in the AdaBoost.MH method, its importance is significantly reduced. AdaBoost.MH is an asymmetric strategy, building separate two class models for each individual class against the pooled complement classes. Pooling classes can produce more complex decision boundaries that are difficult to approximate, while separating class pairs could be relatively simple [58]. Moreover, pseudo-multiclass algorithms might create resource problems by increasing (basically multiplying) *e.g.* training time or memory requirements, especially for problems with a large number of classes. Therefore, to overcome these drawbacks direct multiclass extensions of the AdaBoost method should be developed and investigated. Nevertheless, pseudo-multiclass methods will remain interesting since they can be used for the multiclass multilabeled case: when instances may belong to more than one class [155]. An application scenario for this case is *e.g.* text categorization: one document can be assigned to more than one topic. If the goal is to predict all and only all of the correct labels, the AdaBoost.MH algorithm is a valid solution.

6.2.3 Multiclass Classification

The first direct multiclass extension of the original AdaBoost algorithm, AdaBoost.M1, was introduced in [55] and is the most widely used multiclass boosting method. It is also the basis of many further variants of multiclass boosting. The AdaBoost.M1 algorithm is shown in Algorithm 6.4. Similar to the binary AdaBoost methods, it can be used with any weak classifier that has an error rate of less than 0.5. However, this criterion is more restrictive than for binary classification, where an error rate of 0.5 means basically random guessing. In [55] a second multiclass extension of the original AdaBoost algorithm, AdaBoost.M2, was also introduced. In this algorithm the weak classifiers have to minimize a newly introduced pseudo-loss, instead of minimalizing the error rate as done usually. The pseudo-loss of the weak classifiers has to be less than 0.5, but this is a much weaker condition than the error rate being less than 0.5. The drawback of AdaBoost.M2 is that classifiers have to be redesigned in order to be used as weak learners within this algorithm, since almost all traditionally used classifiers minimize the error rate and not the new pseudo-loss.

In [41, 203] another way to overcome the restriction on the weak learner's error rate is shown by adding a constant taking the number of classes (C) into account, this way relaxing the requirement of the weak classifiers to an error rate of less than random guessing ($1 - \frac{1}{C}$). Eibl and Pfeiffer [41] introduced the AdaBoost.M1W algorithm based on this idea, and proved with experiments its benefits over AdaBoost.M1. The SAMME (Stagewise Additive Modeling using a Multi-class Exponential loss function) algorithm of Zhu et al. [203] is based on the same idea, as shown in Algorithm 6.5. SAMME has the same structure as AdaBoost.M1, the only difference is on line 9 where the term $log(C - 1)$ is added. Zhu et al. [203] show that this extra term is not ar-

Algorithm 6.4 AdaBoost.M1

Require: Training dataset of N instances: $(\underline{x}_i, y_i)$ $i = 1, \ldots, N$ ($\underline{x}_i$: feature vector, $y_i \in [1, \ldots, C]$)
New instance to classify: $\underline{x}_n$

procedure TRAINING($(\underline{x}_i, y_i)$ $i = 1, \ldots, N$)
  Assign equal weight to each training instance: $w_i = \frac{1}{N}$, $i = 1, \ldots, N$
  for $t \leftarrow 1, T$ **do**
    Fit weak learner on the weighted dataset: $f_t(\underline{x}) \in [1, \ldots, C]$
    Compute error e_t of model on weighted dataset: $e_t = \sum_{i:y_i \neq f_t(\underline{x}_i)} w_i$
    if $e_t = 0$ or $e_t \geq 0.5$ **then**
      Delete last $f_t(\underline{x})$ and terminate model generation.
    end if
    Compute $\alpha_t = \log \frac{1-e_t}{e_t}$
    for $i \leftarrow 1, N$ **do**
      if $y_i \neq f_t(\underline{x}_i)$ **then**
        $w_i \leftarrow w_i e^{\alpha_t}$
      end if
    end for
    Normalize the weight of all instances so that $\sum_i w_i = 1$
  end for
end procedure

procedure PREDICTION($\underline{x}_n$)
  Set zero weight to all classes: $\mu_j = 0$, $j = 1, \ldots, C$
  for $t \leftarrow 1, T$ **do**
    Predict class with current model: $c = f_t(\underline{x}_n)$
    $\mu_c \leftarrow \mu_c + \alpha_t$
  end for
  The output class is $\arg\max_j \mu_j \quad j = 1, \ldots, C$
end procedure

tificial: Similar to the interpretation of AdaBoost in [58], SAMME is equivalent to fitting a forward stage-wise additive model using a multiclass exponential loss function. Obviously when $C = 2$, SAMME reduces to AdaBoost.M1. However, the extra term $log(C - 1)$ is critical in the multiclass case, since in order for α_t to be positive only requires $(1 - e_t) > 1/C$. Therefore, the error rate of the weak learners only has to be better than random guessing rather than 0.5. Zhu et al. [203] compared the SAMME algorithm with AdaBoost.MH on various benchmark datasets from the UCI repository. They showed that SAMME's performance is comparable with that of the AdaBoost.MH method, or even slightly better. The SAMME.R variation [202] of the SAMME algorithm uses the probability estimates from the weak classifiers. However, SAMME.R does not keep the structure of AdaBoost.M1: when updating the weights for the training instances only the respective probability estimates are used, the error e_t of the weak learner on the weighted dataset is not considered. Moreover, the

Algorithm 6.5 SAMME

Require: Training dataset of N instances: $(\underline{x}_i, y_i)$ $i = 1, \ldots, N$ ($\underline{x}_i$: feature vector, $y_i \in [1, \ldots, C]$)
New instance to classify: $\underline{x}_n$

procedure TRAINING($(\underline{x}_i, y_i)$ $i = 1, \ldots, N$)
  Assign equal weight to each training instance: $w_i = \frac{1}{N}$, $i = 1, \ldots, N$
  for $t \leftarrow 1, T$ **do**
    Fit weak learner on the weighted dataset: $f_t(\underline{x}) \in [1, \ldots, C]$
    Compute error e_t of model on weighted dataset: $e_t = \sum_{i: y_i \neq f_t(\underline{x}_i)} w_i$
    if $e_t = 0$ or $e_t \geq 1 - \frac{1}{C}$ **then**
      Delete last $f_t(\underline{x})$ and terminate model generation.
    end if
    Compute $\alpha_t = \log \frac{1-e_t}{e_t} + \log(C-1)$
    for $i \leftarrow 1, N$ **do**
      if $y_i \neq f_t(\underline{x}_i)$ **then**
        $w_i \leftarrow w_i e^{\alpha_t}$
      end if
    end for
    Normalize the weight of all instances so that $\sum_i w_i = 1$
  end for
end procedure

procedure PREDICTION($\underline{x}_n$)
  Set zero weight to all classes: $\mu_j = 0$, $j = 1, \ldots, C$
  for $t \leftarrow 1, T$ **do**
    Predict class with current model: $c = f_t(\underline{x}_n)$
    $\mu_c \leftarrow \mu_c + \alpha_t$
  end for
  The output class is $\arg\max_j \mu_j \quad j = 1, \ldots, C$
end procedure

SAMME.R algorithm showed overall slightly worse performance results than SAMME on different datasets [202], thus is discarded from further analysis in this work.

Another multiclass boosting method is introduced in [72]: GAMBLE (Gentle Adaptive Multiclass Boosting Learning) is the generalized version of the binary Gentle AdaBoost algorithm. However, GAMBLE fits a regression model rather than a classification model at each boosting iteration, thus requires several additional steps in order to be used for classification tasks (which is the actual focus of this chapter). First the class labels have to be encoded (*e.g.* with response encoding), then the regression model is fitted which is then used to obtain the weak classifier. Overall, the training time and computational cost is significantly increased compared to AdaBoost models using directly classification models.

Algorithm 6.6 ConfAdaBoost.M1

Require: Training dataset of N instances: $(\underline{x}_i, y_i)$ $i = 1, \ldots, N$ ($\underline{x}_i$: feature vector, $y_i \in [1, \ldots, C]$)
New instance to classify: $\underline{x}_n$

procedure TRAINING($(\underline{x}_i, y_i)$ $i = 1, \ldots, N$)
  Assign equal weight to each training instance: $w_i = \frac{1}{N}$, $i = 1, \ldots, N$
  for $t \leftarrow 1, T$ **do**
    Fit weak learner on the weighted dataset: $f_t(\underline{x}) \in [1, \ldots, C]$
    Compute the confidence of the prediction that instance $\underline{x}_i$ belongs to the predicted class: p_{ti}, $i = 1, \ldots, N$
    Compute error e_t of model on weighted dataset: $e_t = \sum_{i:y_i \neq f_t(\underline{x}_i)} p_{ti} w_i$
    if $e_t = 0$ or $e_t \geq 0.5$ **then**
      Delete last $f_t(\underline{x})$ and terminate model generation.
    end if
    Compute $\alpha_t = \frac{1}{2} \log \frac{1-e_t}{e_t}$
    for $i \leftarrow 1, N$ **do**
      $w_i \leftarrow w_i e^{\left(\frac{1}{2} - \mathbb{I}(y_i = f_t(\underline{x}_i))\right) p_{ti} \alpha_t}$ % $\mathbb{I}()$ *refers to the indicator function*
    end for
    Normalize the weight of all instances so that $\sum_i w_i = 1$
  end for
end procedure

procedure PREDICTION($\underline{x}_n$)
  Set zero weight to all classes: $\mu_j = 0$, $j = 1, \ldots, C$
  for $t \leftarrow 1, T$ **do**
    Predict class with current model:
    $[c, p_t(\underline{x}_n)] = f_t(\underline{x}_n)$, where $p_t(\underline{x}_n)$ is the confidence of the prediction that instance $\underline{x}_n$ belongs to the predicted class c
    $\mu_c \leftarrow \mu_c + p_t(\underline{x}_n) \alpha_t$
  end for
  The output class is $\arg\max_j \mu_j \quad j = 1, \ldots, C$
end procedure

6.3 ConfAdaBoost.M1

Various boosting algorithms exist and were presented in the previous section. However, there are still classification problems where the difficulty of the task exceeds the potential of existing methods. Examples of such complex tasks in the field of physical activity monitoring were shown in the benchmark of [135, 136]. Moreover, experiments presented in this chapter show a high error rate on the PAMAP2 physical activity monitoring dataset with selected, commonly used boosting algorithms. Therefore, there is a need for further development of boosting techniques to improve the performance on such complex classification tasks.

This section introduces a new boosting algorithm, called ConfAdaBoost.M1: A confidence-based extension of the well known AdaBoost.M1 algorithm. The ConfAdaBoost.M1 algorithm is based on the concepts and ideas of previously mentioned boosting methods, and combines some of their benefits. First of all it is a direct multiclass classification technique, thus it overcomes the drawbacks of pseudo-multiclass boosting methods (*cf.* Section 6.2.2). Moreover, it keeps the structure of AdaBoost.M1, thus when already using AdaBoost.M1 in a classification task it can be easily extended to ConfAdaBoost.M1. Furthermore, the new algorithm uses the information about how confident the weak learners are to estimate the class of the instances. This approach has been beneficial in both binary (when developing the Real AdaBoost algorithm from Discrete AdaBoost in [58]) and pseudo-multiclass (the improvement of Discrete AdaBoost.MH to Real AdaBoost.MH in [155]) classification. Therefore, this work takes the next step by applying the idea of a confidence-based version of AdaBoost for the direct multiclass classification case. It is worth to mention that Quinlan [127] already proposed to modify the prediction step of the AdaBoost.M1 algorithm to allow the voting weights of the weak learners to vary in response to the confidence with which $\underline{x}_n$ (the new instance) is classified. However, no confidence-based extension of the training part of the AdaBoost.M1 algorithm has previously been proposed.

The main idea of the ConfAdaBoost.M1 algorithm can be described as follows. In the training part of the algorithm the confidence of the classification estimation is returned for each instance by the weak learner, and is then used to compute the new weight of that instance: the more confident the weak learner is in a correct classification the more the weight will be reduced, and the more confident the weak learner is in a misclassification the more the weight will be increased. Moreover, the confidence values are also used in the prediction part of the algorithm: The more confident the weak learner is in a new instance's prediction the more it counts in the output of the combined classifier, as proposed in [127].

The ConfAdaBoost.M1 algorithm is shown in Algorithm 6.6. The structure of the original AdaBoost.M1 algorithm is kept (*cf.* Algorithm 6.4), extending it on multiple lines. First of all, after training the weak learner on the weighted dataset (line 4), the confidence of the classification estimation is returned for each instance by this weak learner (line 5). These p_{ti} confidence values are used when computing the error rate of the weak learner (line 6): the more confident the model is in the misclassification the more that instance's weight counts in the overall error rate. The factor $\frac{1}{2}$ on line 10 of the ConfAdaBoost.M1 algorithm is used to compensate the lower e_t compared to the computed error rate of AdaBoost.M1. The p_{ti} confidence values are also used to recomputing the weights of the instances. The more confident the weak learner is in an instance's correct classification or misclassification, the more that instance's weight is reduced or increased, respectively (line 12). The factor $\frac{1}{2}$ on line 12 (determined in an empirical study) is applied in addition compared to the original AdaBoost.M1 algorithm, to compensate that weights are modified in both directions before the renormalization of the weights. In the prediction part of ConfAdaBoost.M1 the only modification compared to the AdaBoost.M1 algorithm is that the confidence of the prediction ($p_t(\underline{x}_n)$) is computed (line 20), and then used to adjust the voting weights of the weak learners (line 21).

It should be noted that the stopping criterion of $e_t \geq 0.5$ in the original AdaBoost.M1 remains the same in the new ConfAdaBoost.M1 algorithm (line 7 of Algorithm 6.6). This means that, similar to AdaBoost.M1, only classifiers achieving a reasonably high accuracy value can be used as weak learners, thus *e.g.* decision stumps are not suitable for multiclass problems. However, the stopping criterion of $e_t \geq 0.5$ is less restrictive in ConfAdaBoost.M1, since the computation of the error rate also uses the p_{ti} confidence values, thus the computed e_t is lower than in the original AdaBoost.M1 algorithm. Therefore, when using the same weak learner, ConfAdaBoost.M1 can perform significantly more boosting iterations before stopping compared to AdaBoost.M1, as shown in the experiments of the next sections.

6.4 Evaluation on UCI Datasets

In this section experiments on various datasets from the UCI machine learning repository [12] are presented. These experiments compare the newly introduced ConfAdaBoost.M1 algorithm to the most commonly used existing boosting methods. The first part of this section presents the basic conditions of the experiments, then results are given and discussed.

6.4.1 Basic Conditions

The experiments were performed on 8 datasets from the UCI repository. The selected benchmark datasets include 3 small datasets: *Glass*, *Iris* [49] and *Vehicle* [162], as well as 5 pre-partitioned larger datasets: *Letter* [57], *Pendigits* [6], *Satimage*, *Segmentation* and *Thyroid* [128]. The parameters of the used datasets are summarized in Table 6.1. These datasets were selected with the goal to cover a wide range of scenarios: The size of the datasets ranges from 150 to 20 000 instances, the number of classes ranges from 3 to 26, and the difficulty of the classification problems these datasets define vary a lot as well according to experiments performed on these datasets in previous work (*cf. e.g.* [155, 203]). A further selection criterion was to only include datasets which directly provide the features of the classification tasks as attributes, thus no domain knowledge (*e.g.* how to process the provided data, which features should be extracted, *etc.*) of the datasets should be required. Using various datasets from the UCI repository is common practice when introducing a new boosting method and comparing it to existing algorithms. For instance, Zhu et al. [203] used 7 different UCI datasets to compare their SAMME algorithm to AdaBoost.MH, and Jin et al. [81] used 23 UCI datasets to compare their proposed AdaBoost.HM algorithm to AdaBoost.M1 and AdaBoost.MH. Finally, using datasets which have been applied before allows a real comparison to previous work.

On the selected datasets, the ConfAdaBoost.M1 algorithm is compared to 4 other existing boosting methods. First of all to AdaBoost.M1 to provide the baseline performance of the experiments (since, as many other boosting variants, ConfAdaBoost.M1 is also an extension of AdaBoost.M1). The proposed confidence-based modification of the prediction step of AdaBoost.M1 in [127] is part of the ConfAdaBoost.M1 algorithm. Therefore, it is of interest to compare ConfAdaBoost.M1 to this extension

Table 6.1: *Summary of the benchmark datasets used in the experiments*

Dataset	#Instances			#Variables	#Classes
	Total	Training	Testing		
Glass	214	—	—	9	6
Iris	150	—	—	4	3
Vehicle	846	—	—	18	4
Letter	20000	16000	4000	16	26
Pendigits	10992	7494	3498	16	10
Satimage	6435	4435	2000	36	6
Segmentation	2310	210	2100	19	7
Thyroid	7200	3772	3428	21	3
PAMAP2_AR	19863	—	—	137	15
PAMAP2_IE	24197	—	—	137	3

of the original AdaBoost.M1, to investigate whether possible performance improvements come from only the confidence-based prediction step or the confidence-based extension of both the training and prediction steps, as proposed by ConfAdaBoost.M1. The confidence-based modification of the prediction step by Quinlan [127] will be referred to as QuinlanAdaBoost.M1 hereafter: this algorithm is constructed from the training step of the original AdaBoost.M1 algorithm and the prediction step of the ConfAdaBoost.M1 algorithm. The QuinlanAdaBoost.M1 algorithm is to be expected slightly better than the original AdaBoost.M1, according to [127]. The next boosting method used for comparison is SAMME (*cf.* Algorithm 6.5), since according to [41, 203] this direct multiclass extension of AdaBoost.M1 outperforms traditionally used boosting techniques. Finally, the most common pseudo-multiclass classification technique is used for comparison: the Real AdaBoost.MH algorithm (*cf.* Algorithm 6.3). It performs best amongst the pseudo-multiclass methods and is a confidence-based boosting version similar to ConfAdaBoost.M1.

The C4.5 decision tree classifier [126] is used as weak learner in each of the evaluated boosting methods. This classifier is, together with decision stumps, the most commonly used weak learner for boosting. It also fulfills the requirement of achieving a reasonably high accuracy on the different classification problems (it has an error rate of significantly less than 0.5 on the various datasets, as shown below by the results), thus can be used with the algorithms AdaBoost.M1, QuinlanAdaBoost.M1 and ConfAdaBoost.M1. Considering confidence-based versions of AdaBoost, the C4.5 decision tree has another benefit: there is no need to modify the C4.5 algorithm, the confidence values of the weak learners' predictions can be directly extracted from the trained decision trees. Assume that a C4.5 decision tree is trained as $f_t(\underline{x})$ weak learner in the ConfAdaBoost.M1 algorithm (Algorithm 6.6, line 4). The p_{ti} confidence of the prediction that instance $\underline{x}_i$ belongs to the predicted class (Algorithm 6.6, line 5) can be computed as follows, based on [127]. In the trained C4.5 decision tree a single leaf node classifies $\underline{x}_i$: $c = f_t(\underline{x}_i)$. Let S be the training instances mapped to this leaf,

and let S_c be the subset of S belonging to class c. The confidence of the prediction is then:

$$p_{ti} = \sum_{j \in S_c} w_j \Big/ \sum_{j \in S} w_j. \tag{6.1}$$

On the 5 larger, pre-partitioned datasets pruned C4.5 decision trees are used. The level of pruning is defined by 5-fold cross-validation (CV) on the training part of these datasets, for each of the evaluated boosting methods separately. On the 3 smaller datasets (Glass, Iris and Vehicle), non-pruned C4.5 decision trees are used as weak learners. Between 1 and 500 boosting iterations are evaluated for all algorithms and benchmark datasets (previous work *e.g.* in [41, 203] showed that the performance of various boosting algorithms usually levels off at maximum 100 iterations). All results presented below are averages of multiple test runs. On datasets providing a training and test part training is performed 10 times on the training set, and the trained classifier is then evaluated on the provided test set each time. On datasets without a predefined test part, 10-fold CV is used and performed 10 separate times. All experiments were performed within Matlab, random substreams are used to ensure randomness between different test runs.

6.4.2 Results and Discussion

The averaged results of the 10 test runs on the selected 8 UCI benchmark datasets are shown in Figure 6.1 and Figure 6.2. The test errors of the 5 evaluated boosting methods are summarized in Table 6.2. Overall it is clear that the ConfAdaBoost.M1 algorithm performed best in the experiments: on 7 out of 8 datasets there is a noticeable increase in performance compared to existing boosting methods, while on one dataset (Thyroid) ConfAdaBoost.M1 has essentially the same performance as the other algorithms. According to the results of Table 6.2, the second best boosting algorithm is SAMME, closely followed by AdaBoost.MH, confirming the results of [203]. The original AdaBoost.M1 and its variation QuinlanAdaBoost.M1 performed overall clearly worse, the latter algorithm being slightly but not significantly better.

A statistical significance test (the McNemar test [85] is used to pair-wise compare the predictions of the different methods) indicates that the reduction of the test error rate by ConfAdaBoost.M1 compared to SAMME is significant with p-value 0.01 on the datasets Pendigits and Segmentation, significant with p-value 0.05 on the datasets Letter and Satimage, and that on the remaining datasets no statistical significance was observed. In conclusion, the ConfAdaBoost.M1 algorithm has more potential for improvement the larger the dataset and the more complex the classification problem is. This statement is supported by the results on the PAMAP2 classification tasks in the next section. Moreover, similar observation was made by Schapire and Singer [155] when comparing Discrete and Real AdaBoost.MH: the confidence-based method had better capability for improvement the larger the datasets were. On the Thyroid dataset on the other hand even AdaBoost.M1 reaches an accuracy of over 99% leaving only a few outlier instances misclassified, thus explaining the minimal (not statistically significant) difference between the results of the 5 algorithms. Furthermore, Friedman et al. [58] conclude that interpreting results and slight performance differences on

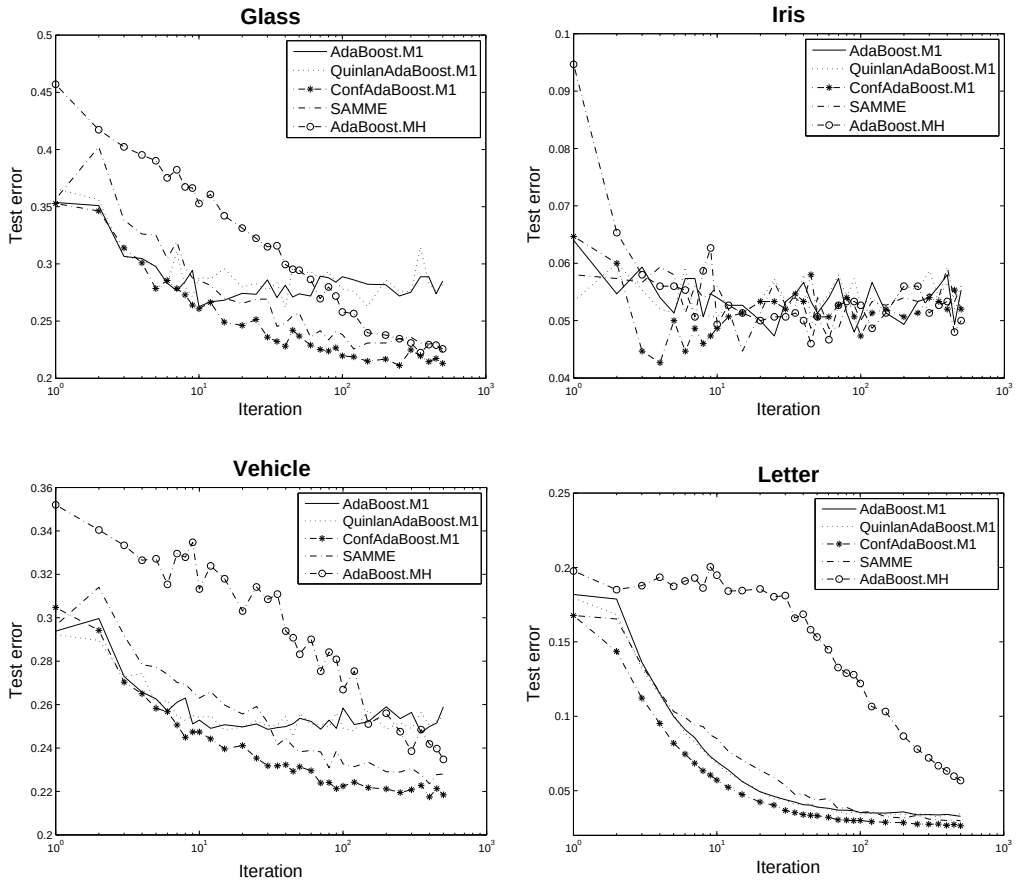


Figure 6.1: *Test error of the* 5 *evaluated boosting algorithms on the UCI benchmark datasets Glass, Iris, Vehicle and Letter. The results are averages over* 10 *test runs.*

rather small datasets is difficult since it can occur due to sampling fluctuations, while on the larger datasets clearer trends are observable.

One of the main reasons why AdaBoost.M1 and QuinlanAdaBoost.M1 performs significantly worse than the other methods is that they reach the stopping criterion of $e_t \geq 0.5$ quickly. This can be observed especially on the results of the datasets Glass, Vehicle or Satimage: the test error decreases at the beginning but levels off already at around 10 to 20 boosting iterations, no further improvement can be reached with the increase of the number of boosting rounds. This effect is not observed when using the ConfAdaBoost.M1 algorithm due to the modified computation of the error rate of the weak learners. Another benefit of ConfAdaBoost.M1 over the other methods can be observed *e.g.* on the results of the datasets Vehicle, Letter and Satimage: the test error even at lower numbers of boosting iterations is the lowest when using ConfAdaBoost.M1. This means that for a particular level of accuracy fewer boosting rounds are necessary with ConfAdaBoost.M1, thus a smaller classifier size is required for the same performance compared to existing boosting algorithms. This quality is

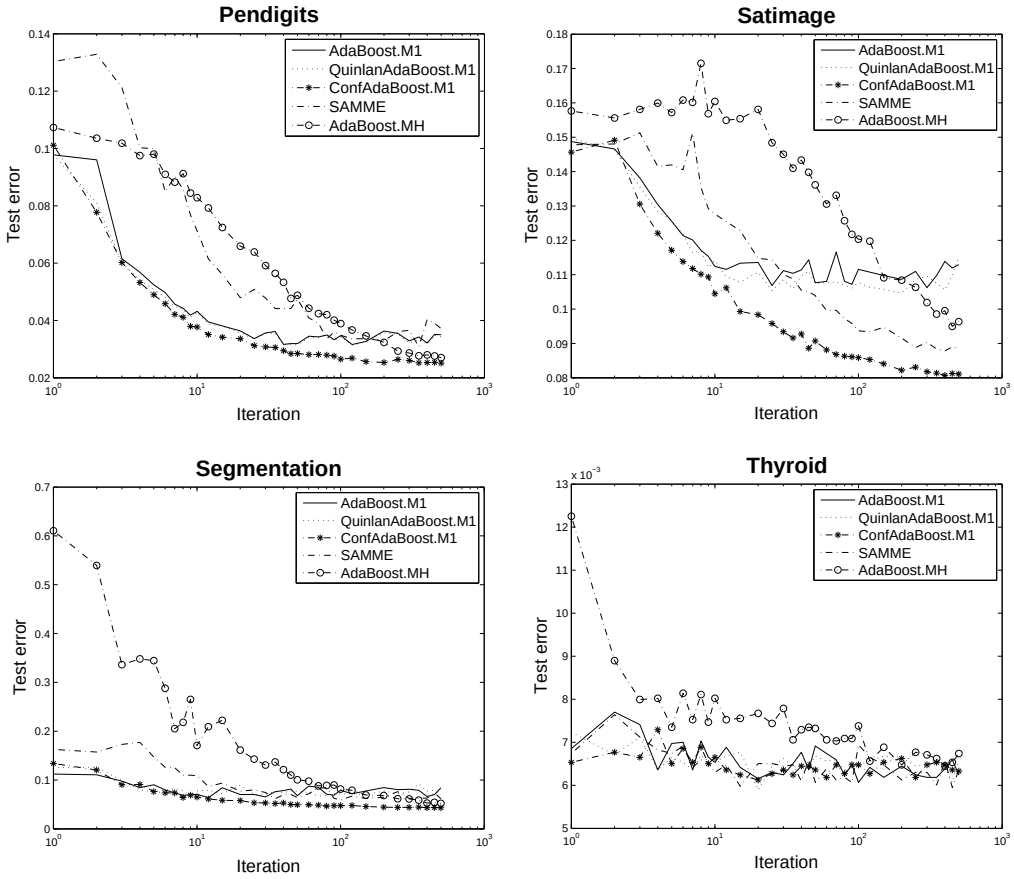


Figure 6.2: *Test error of the* 5 *evaluated boosting algorithms on the UCI benchmark datasets Pendigits, Satimage, Segmentation and Thyroid. The results are averages over* 10 *test runs.*

especially beneficial when the available computational resources are limited, which is usually the case for physical activity monitoring applications.

Finally, it is worth to discuss and compare the training time required for creating the different classifiers. Building a decision tree has the time complexity of $\mathcal{O}(DMN\log(N))$, where N is the number of training instances, M is the dimension of the feature vector of a training instance, and D is the average depth of the decision tree [202]. The computational cost of AdaBoost.M1 is then $\mathcal{O}(DMN\log(N)T)$, where T is the number of boosting iterations. The theoretical complexity of the algorithms QuinlanAdaBoost.M1, SAMME and the newly proposed ConfAdaBoost.M1 is similar. The computational cost of AdaBoost.MH is $\mathcal{O}(DMN\log(N)TC)$, where C refers to the number of classes. During the experiments of this section, the training time of ConfAdaBoost.M1 was comparable to that of SAMME on all 8 evaluated datasets. Compared to these two algorithms, the training time of AdaBoost.M1 and QuinlanAdaBoost.M1 was almost an order of magnitude lower. This can be explained with the early reaching of the stopping criterion, as discussed in the previous paragraph (thus T gets smaller in the expression of $\mathcal{O}(DMN\log(N)T)$). On the other

Table 6.2: *Comparison of the* 5 *evaluated boosting algorithms: test error rates [%] on the selected benchmark datasets. The results are averaged over* 10 *test runs (mean and standard deviation are given), the best performance is shown for each of the methods.*

Dataset	**AdaBoost.M1**	**Quinlan-AdaBoost.M1**	**Conf-AdaBoost.M1**	**SAMME**	**AdaBoost.MH**
Glass	26.26 ± 1.42	26.17 ± 2.60	**21.12** ± 1.22	22.29 ± 1.38	22.24 ± 1.81
Iris	4.73 ± 0.73	5.00 ± 0.85	**4.27** ± 0.64	4.47 ± 1.22	4.60 ± 0.80
Vehicle	24.72 ± 1.05	24.52 ± 1.10	**21.75** ± 0.44	22.35 ± 1.14	23.48 ± 1.21
Letter	3.28 ± 0.14	3.19 ± 0.15	**2.64** ± 0.11	2.99 ± 0.13	5.68 ± 0.39
Pendigits	3.16 ± 0.27	3.14 ± 0.45	**2.51** ± 0.11	3.08 ± 0.15	2.70 ± 0.08
Satimage	10.63 ± 0.80	10.47 ± 1.01	**8.07** ± 0.15	8.79 ± 0.25	9.50 ± 0.39
Segmentation	6.36 ± 1.03	6.55 ± 1.08	**4.31** ± 0.20	5.92 ± 0.79	5.22 ± 0.78
Thyroid	0.61 ± 0.04	**0.59** ± 0.05	0.61 ± 0.05	0.60 ± 0.06	0.64 ± 0.08
PAMAP2_AR	29.28 ± 1.40	27.90 ± 1.06	**22.22** ± 0.77	27.98 ± 1.34	—
PAMAP2_IE	7.98 ± 1.04	7.73 ± 0.66	**5.60** ± 0.31	7.81 ± 0.60	—

hand, the training time required for AdaBoost.MH was 20 to 40 times larger than for ConfAdaBoost.M1 on the larger datasets (*e.g.* Letter or Pendigits). Therefore, training AdaBoost.MH is not feasible for extremely large datasets.

6.5 Evaluation on the PAMAP2 Dataset

The PAMAP2 dataset is a physical activity monitoring dataset created and released recently [135, 136], and is included in the UCI machine learning repository as well. The dataset was recorded from 18 physical activities performed by 9 subjects, wearing 3 inertial measurement units (IMU) and a heart rate monitor. Each of the subjects followed a predefined data collection protocol of 12 activities (lie, sit, stand, walk, run, cycle, Nordic walk, iron, vacuum clean, rope jump, ascend and descend stairs), and optionally performed a few other activities (watch TV, computer work, drive car, fold laundry, clean house, play soccer). Therefore, the PAMAP2 dataset not only includes basic physical activities and postures, but also a wide range of everyday, household and fitness activities. A more detailed description of the dataset can be found in Section 3.3.

In this section first an activity recognition and an intensity estimation classification problem is defined on the PAMAP2 dataset. The reason for defining these classification tasks is to show that ConfAdaBoost.M1 performs well on both main objectives of this thesis, namely both on activity recognition and on intensity estimation (*cf.* Section 1.2). The two defined classification problems are described in detail, highlighting also the differences to the UCI benchmark datasets of the previous section and pointing out the special challenge these problems pose. Using the defined classification tasks different boosting methods are evaluated and compared to the proposed ConfAdaBoost.M1 algorithm.

6.5.1 Definition of the Classification Problems

The benchmark of Section 4.4 defined 4 different classification problems on the PAMAP2 dataset. One of these problems – called *All activity recognition task* – uses the 12 activities of the data collection protocol, defining 12 classes corresponding to the activities. This classification task is extended in this section with 3 additional activities from the optional activity list: fold laundry, clean house and play soccer.[3] This activity recognition task of 15 different activity classes will be referred to as the 'PAMAP2_AR' task throughout this chapter. Moreover, an intensity estimation classification task is defined on the PAMAP2 dataset: using all 18 activities, the goal is to distinguish activities of light, moderate and vigorous effort (referred to as 'PAMAP2_IE' task). The ground truth for this rough intensity estimation task is based on the metabolic equivalent (MET) of the different physical activities, provided by [1]. Therefore, the 3 intensity classes are defined as follows: lie, sit, stand, drive car, iron, fold laundry, clean house, watch TV and computer work are regarded as activities of

[3]The remaining 3 activities from the dataset are discarded from the activity recognition task for the following reasons: *drive car* contains data from only one subject, while *watch TV* and *computer work* are not considered due to their high resemblance to the *sit* class.

light effort (< 3.0 METs); walk, cycle, descend stairs, vacuum clean and Nordic walk as activities of moderate effort (3.0-6.0 METs); run, ascend stairs, rope jump and play soccer as activities of vigorous effort (> 6.0 METs).

Contrary to the 8 UCI benchmark datasets used for the experiments in the previous section, the PAMAP2 dataset does not directly provide a feature vector with each of the instances, but provides only raw sensory data from the 3 IMUs and the heart rate monitor. Therefore, the raw signal data needs to be processed first in order to be used by classification algorithms. A data processing chain is applied on the raw sensory data including preprocessing, segmentation and feature extraction steps (these data processing steps are further described in Section 4.2). In total, 137 features are extracted: 133 features from IMU acceleration data (such as mean, standard deviation, energy, entropy, correlation, etc.) and 4 features from heart rate data (mean and gradient). These extracted features serve as input to the classification step, in which different boosting algorithms are evaluated. The main parameters of the PAMAP2 classification tasks are summarized in Table 6.1. It is clear that, compared to the other datasets of Table 6.1, the classification problems defined on the PAMAP2 dataset are significantly more complex, considering the number of instances and especially the number of variables. To get a first impression about the difficulty of these tasks, experiments with a C4.5 decision tree classifier are performed: 65.79% is reached on the PAMAP2_AR and 88.98% on the PAMAP2_IE task, averaged over 10 test runs. This result serves as baseline performance, showing that improvement is required and to be expected while applying different boosting methods.

The experiments presented below in this section compare the newly introduced ConfAdaBoost.M1 algorithm to the boosting methods AdaBoost.M1, QuinlanAdaBoost.M1 and SAMME. The selection of these algorithms for comparison was already explained in Section 6.4.1. The comparison to AdaBoost.MH is not considered here due to the unfeasible training time it would require, given the complexity of the classification tasks and that the actual size of the training set is a multiple of that of the other algorithms (*cf.* also the discussion in Section 6.4.2). Similar to the previous section, the C4.5 decision tree classifier is used for each of the boosting algorithms as weak learner. An important difference in the realization of the experiments in this section is the applied evaluation technique. As discussed in Section 5.1.2, a subject independent validation technique simulates best the goals of systems and applications using physical activity recognition. Therefore, leave-one-subject-out (LOSO) 9-fold cross-validation is used in this section, while evaluating each method from 1 up to 500 boosting iterations.

6.5.2 Results and Discussion

The averaged results of the 10 test runs on the PAMAP2_AR classification task are shown in Figure 6.3, and on the PAMAP2_IE task in Figure 6.4, respectively. The test error rates of the 4 evaluated boosting methods are included in Table 6.2. Compared to the baseline accuracy of the decision tree classifier, all boosting methods significantly improve the performance. The ConfAdaBoost.M1 algorithm clearly outperforms the other methods: *e.g.* on the PAMAP2_AR task, compared to the perfor-

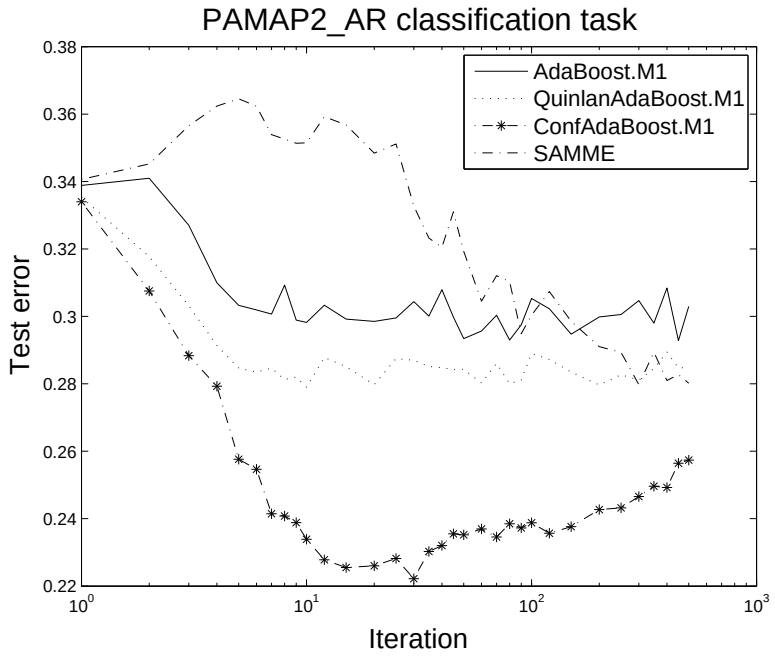


Figure 6.3: *Test error of the* 4 *evaluated boosting algorithms on the PAMAP2_AR classification task. The results are averages over* 10 *test runs.*

mance of the second best SAMME algorithm a reduction of the test error rate by nearly 20% can be observed. This reduction of the test error rate is statistically significant with a p-value smaller than 0.001. As discussed in Section 6.4.2, it was expected that the most significant improvement from all the datasets evaluated in this chapter is achieved on the PAMAP2_AR classification task, since it represents the largest and most complex classification problem.

Similar to the results of Figure 6.1 and Figure 6.2, the algorithms AdaBoost.M1 and QuinlanAdaBoost.M1 reach the stopping criterion at lower boosting iteration numbers. However, contrary to the results of the previous section, QuinlanAdaBoost.M1 performs significantly better here (especially on the PAMAP2_AR task), confirming that it is even worth to apply the confidence-based modification to only the prediction step of the original AdaBoost.M1 algorithm, as proposed in [127]. However, compared to QuinlanAdaBoost.M1, ConfAdaBoost.M1 reduces the test error rate by 20%. Therefore, the major part of the performance improvement achieved by ConfAdaBoost.M1 comes from the confidence-based extension of both the training and prediction step of the original AdaBoost.M1 algorithm, as also confirmed by the results on the 8 other UCI datasets. Therefore ConfAdaBoost.M1 is clearly a significant improvement over QuinlanAdaBoost.M1.

The typical behaviour of boosting in respect of increasing the number of boosting iterations shows the following scheme: the performance increases and levels off at a certain number of boosting rounds, by further increasing the iteration number the performance remains at the maximum level and does not decrease, thus boosting is usually resistant to overfitting. This behaviour of boosting was the topic of many research

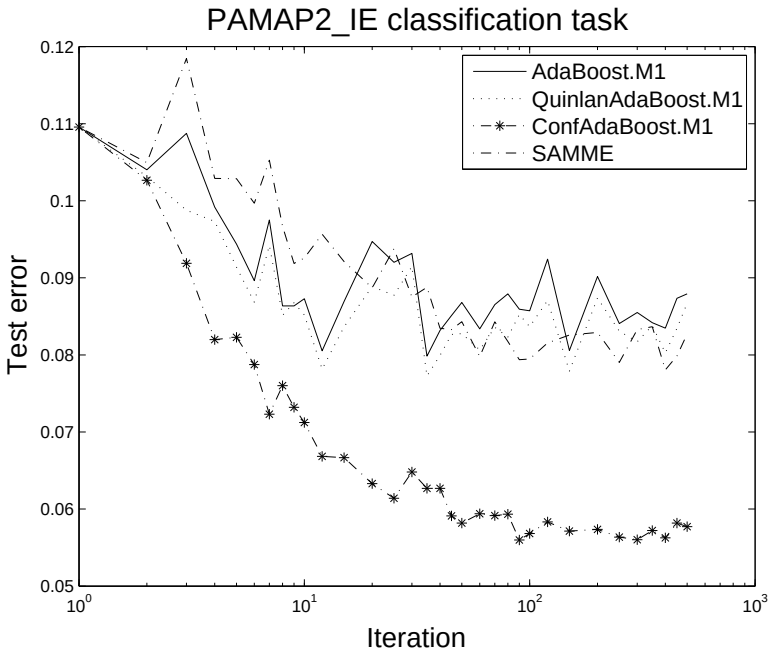


Figure 6.4: *Test error of the* 4 *evaluated boosting algorithms on the PAMAP2_IE classification task. The results are averages over* 10 *test runs.*

work in the past (*e.g.* in [56, 58, 110]), only a limited number of examples is known where overfitting with boosting occurs. All the results presented on the various UCI datasets show this advantageous behaviour. ConfAdaBoost.M1 adopts this beneficial characteristics of boosting: it rarely overfits a classification problem. The only result indicating overfitting is on the PAMAP2_AR task (*cf.* Figure 6.3): after decreasing the test error and reaching the best performance at 30 boosting rounds, the test error slightly increases again with increasing numbers of boosting iterations. It is an interesting question why overfitting occurs here, and why only on the PAMAP2_AR task with only the ConfAdaBoost.M1 method, which needs further investigation. Nevertheless, even with higher numbers of boosting iterations (*e.g.* with 500 boosting rounds) the performance of ConfAdaBoost.M1 is significantly better than that of the other evaluated boosting methods.

To better understand the results of this section, the confusion matrix of the best performing classifier (ConfAdaBoost.M1 with 30 boosting iterations) on the PAMAP2_AR task is presented in Table 6.3. The numbering of the activities in the table corresponds to the activity IDs as given in the PAMAP2 dataset. The results are averaged over 10 test runs, the overall accuracy is 77.78%. The confusion matrix shows that some activities are recognized with high accuracy, *e.g.* lie, walk or even distinguishing between ascend and descend stairs. Misclassifications in Table 6.3 have several reasons. For example, the over 5% confusion between sit and stand can be explained with the positioning of the sensors: an IMU on the thigh would be needed for a reliable differentiation of these postures. Moreover, ironing has a similar characteristics from the used set of sensors' point of view, especially compared to

Table 6.3: *Confusion matrix of the PAMAP2_AR classification task using the ConfAdaBoost.M1 classifier and 30 boosting iterations. The table shows how different annotated activities are classified in [%].*

Annotated	**Recognized activity**														
activity	1	2	3	4	5	6	7	12	13	16	17	18	19	20	24
1 lie	97.1	1.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.0	0.0
2 sit	2.0	84.8	5.4	0.0	0.0	0.5	0.0	0.0	0.0	0.1	4.1	0.6	2.5	0.0	0.0
3 stand	0.0	6.0	83.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	7.4	0.9	2.4	0.0	0.0
4 walk	0.0	0.0	0.0	92.2	0.0	0.0	0.5	6.8	0.0	0.0	0.0	0.0	0.0	0.4	0.0
5 run	0.0	0.0	0.0	0.0	89.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.1	0.0
6 cycle	0.0	0.0	0.0	1.1	0.0	91.7	0.4	0.5	0.0	1.3	0.1	0.0	4.9	0.0	0.0
7 Nordic walk	0.0	0.0	0.0	2.7	0.0	0.0	89.1	1.1	0.1	0.0	0.0	0.0	0.1	7.0	0.0
12 asc. stairs	0.0	0.0	0.0	6.4	0.0	0.2	0.3	87.2	2.6	0.7	0.0	0.0	0.3	2.5	0.0
13 desc. stairs	0.0	0.0	0.0	0.1	0.1	0.0	0.2	6.7	91.3	0.0	0.0	0.0	0.2	0.7	0.7
16 vacuum clean	0.0	0.0	0.1	0.0	0.0	1.1	0.0	0.3	0.4	73.5	1.3	0.3	23.1	0.0	0.0
17 iron	0.0	2.6	0.8	0.0	0.0	0.0	0.0	0.0	0.0	1.2	77.7	5.0	12.7	0.0	0.0
18 fold laundry	0.0	1.1	1.5	0.0	0.0	0.1	0.0	0.0	0.0	8.9	61.1	11.1	16.2	0.0	0.0
19 clean house	0.5	0.6	3.4	0.0	0.0	1.7	0.0	1.2	0.7	21.4	18.1	5.0	47.4	0.0	0.0
20 play soccer	0.0	0.0	0.0	5.1	27.6	1.4	2.8	7.3	20.6	1.7	0.0	0.0	0.1	20.7	12.8
24 rope jump	0.0	0.0	0.0	0.0	32.0	0.0	0.0	1.3	0.1	0.0	0.0	0.0	0.0	7.8	58.8

talking and gesticulating during standing. Another example of overlapping activity characteristics comes from the introduction of playing soccer into this classification problem. Playing soccer is a composite activity, and it is for instance not trivial to distinguish running with a ball from just running. The significant confusion between the different household activities (vacuum clean, iron, fold laundry and clean house – the latter mainly consisting of dusting shelves) indicates that they can not be reliably distinguished with the given set of sensors. However, arguably, the main reason for the misclassifications in Table 6.3 is the diversity in how subjects perform physical activities. Therefore, to further increase the accuracy of physical activity recognition, personalization approaches should be introduced and investigated.

6.6 Conclusion

This chapter introduced a confidence-based extension of the well-known AdaBoost.M1 algorithm, called ConfAdaBoost.M1. The new algorithm builds on established ideas of existing boosting methods, combining some of their benefits. The ConfAdaBoost.M1 algorithm has been evaluated on various benchmark datasets, comparing it to the most commonly used boosting techniques. ConfAdaBoost.M1 performed significantly best among these algorithms, especially on the larger and more complex physical activity monitoring problems: on the PAMAP2_AR task the test error rate was reduced by nearly 20% compared to the second best performing classifier. Therefore, the main motivation of proposing this new boosting variant – namely to overcome some of the challenges defined by recent benchmark results in physical activity monitoring – was achieved successfully.

This chapter presented experimental proof on various datasets in different application areas that the ConfAdaBoost.M1 algorithm is superior to existing methods, and using it improves on classification performance. The main concepts of the new method are clear and comprehensible, but a theoretical interpretation of the algorithm and explanation of its success remains for future work. Moreover, it is also planned to slightly modify ConfAdaBoost.M1 – similar to *e.g.* the modification proposed by SAMME over the original AdaBoost.M1 algorithm – to loosen the stopping criterion of $e_t \geq 0.5$, thus allowing the usage of "weak" weak learners (such as decision stumps). However, boosting decision trees proved to be very successful in the experiments presented in this chapter, and will remain (due to its many benefits discussed above) one of the most widely used classifiers especially in the field of physical activity monitoring.

7

Personalization of Physical Activity Recognition

7.1 Introduction

The previous chapter introduced a novel classification algorithm with the goal to increase the performance on physical activity monitoring tasks. Although the achieved results were very promising, even the best performing classifier only achieved an overall accuracy of 77.78% on the defined complex activity recognition task. The discussion of the respective confusion matrix given in Table 6.3 revealed that the main reason of the remaining confusion between different activities is the diversity in how individuals perform these activities. Therefore, to further increase the accuracy of physical activity recognition[1], this chapter introduces and investigates personalization approaches.

7.1.1 Related Work

Personalization of physical activity recognition has become a topic of interest recently. These approaches are motivated by the fact that activity recognition systems are usually trained on a large number of subjects, and then used by a new subject from whom data is not available in the training phase. Further motivation is given due to the high variance of per-subject activity recognition rates, reported in different previous works. For example, the benchmark results on the recently released PAMAP2 physical activity monitoring dataset (*cf.* Section 4.4) show that although very good overall performance is achieved on various activity recognition tasks, the individual performance of the included subjects varies a lot. Similarly, Weiss and Lockhart [190]

[1] Results presented in Section 6.5.2 show good performance on the defined intensity estimation task, the ConfAdaBoost.M1 classifier achieved an overall accuracy of 94.40%. Therefore, from the two main goals of physical activity monitoring systems, this chapter focuses on the task of activity recognition. However, the here presented approaches can be also applied on the intensity estimation classification task in a trivial way.

also show that the per-user accuracy can have a large variance, resulting that the performance for some users is very poor.

There exist personalization approaches focusing on the feature extraction step of the activity recognition chain (ARC, defined in [145]), *e.g.* by using normalized heart rate, where normalization is done with personal information such as age or resting heart rate (*cf.* [173] or Section 4.2.3). These personalized features proved to be more valuable than absolute (unnormalized) features: They are preferably selected in the decision nodes of trained decision tree-based classifiers, as shown in the preliminary studies of Section 4.2.4.

However, most approaches focus on the classification step of the ARC. A common personalization concept is to adapt the parameters of a previously trained general model to the new user. For example, Pärkkä et al. [120] create a custom decision tree for the recognition of 5 basic activities, and change the thresholds of the decision nodes based on labeled data from the new user. In [201] the parameters of a decision tree are updated using the K-means algorithm with unlabeled data from the previously unknown subject. Furthermore, Berchtold et al. [16] use fuzzy inference system: the new user has to record $1 - 3$ minutes from each activity the system recognizes, and with this data first the best classifier is selected from a set of classifiers, and then adapted to the new user's data.

The drawback of changing the parameters of a general model is that either the model is simple (*e.g.* the decision tree classifiers in [120, 201]) and thus only low performance can be expected on more challenging activity recognition tasks, or the general model is complex and thus resulting in unfeasible computational costs for mobile applications. Another personalization concept is presented in [105]: based on the physical characteristics of the new user a subset of users is selected from a dataset of 40 subjects, and only this subset is used to model the physical activities of the new user. Drawbacks of this approach are that a very large original dataset is required to cover all different types of users, and no significant difference is shown between selecting users based on their physical characteristics and random selection. The reason is that there is not necessarily a high correlation between the physical characteristics of subjects and their movement patterns. Therefore, it is more promising to directly use activity data for the personalization of a general model.

7.1.2 Problem Statement and Contributions

The main focus of this chapter is on the personalization of physical activity recognition, concretely for mobile applications. Considering also the requirements of mobile systems, the following specifications are defined. Since the computational resources of mobile systems (*cf. e.g.* smartphone-based applications) are limited, a computationally not intensive approach is required. Moreover, it is expected that the user (after recording new data) receives the personalized model within a short time. Another requirement is that the personalization concept can handle complex activity recognition tasks (*e.g.* the recognition of not only a few basic, but a large number of physical activities), thus the personalization of advanced classifiers should be feasible. Moreover, contrary to most existing personalization approaches, the new user should not be required to record data from all activities the system recognizes.

This chapter presents a novel general concept of personalization fulfilling the above criteria: personalization is applied in the decision fusion step of the ARC. In this concept the general model consists of a set of classifiers (experts) all weighted the same. Using new labeled data from a previously unknown subject, only the weights of the experts are retrained, the classifiers themselves remain the same. One of the main contributions of this work is to show that this concept is a valid approach for personalization: Different methods based on the idea of weighted majority voting are successfully applied to increase the performance of the general model for new individuals. The second main contribution is the introduction of a new algorithm based on the above concept. In the experiments of this chapter, data from a new user is given by recording a certain amount of labeled data from the different activities, as done in the above presented related work of personalization approaches. However, it is worth to note that both the novel general concept and the new algorithm can also be combined with semi-supervised methods (personalization through semi-supervised learning is presented *e.g.* by Cvetković et al. [37]).

The rest of this chapter is organized in the following way: Section 7.2 presents different methods to retrain the general model, including various weighted majority voting based approaches and a novel algorithm. Section 7.3 first describes the basic conditions of the experiments, then results are presented and discussed. Section 7.4 shows the feasibility of the proposed approaches for mobile activity recognition applications. Finally, the chapter is summarized in Section 7.5.

7.2 Algorithms

This section presents the different algorithms used for the experiments. The general model consists of a set of S classifiers, created from the original training data. In this chapter a single classifier corresponds to a single subject from the training dataset. However, both the new concept of applying personalization in the decision fusion step of the ARC and the novel algorithm based on this concept, can be used with any set of classifiers if there is high variance between their training data.

Each classifier has the same weight in the general model: $w_i = 1$, $i = 1, \ldots, S$. Section 7.2.1 presents different methods based on weighted majority voting, which can be applied to retrain the weights of the classifiers. Moreover, Section 7.2.2 introduces a novel algorithm to retrain the weights using new labeled samples. The baseline performance for these approaches is given by Majority Voting (MV), thus when no retraining of the weights is performed. For a new data instance to be classified each of the equally weighted classifiers of the general model gives a prediction, and the returning activity class is the one with the highest overall accumulated weight (in case of multiple classes having the same highest weight a random selection is made).

7.2.1 Weighted Majority Voting

Given the set of S classifiers, this ensemble learner is personalized with a set of N labeled samples from the new subject, modifying the w_i weights. Several approaches exist which can be applied for this general concept. Since there is no prior informa-

tion about how well the experts perform on the new subject's data, no assumptions can be made about the quality of predicting the previously unknown subject's activity labels. However, the below presented methods follow the natural goal to perform at least nearly as well as the best expert of the general model would.

The first approach which will be used in the experiments of this chapter is the Weighted Majority Algorithm (WMA), described by Blum [20]. In WMA, for each of the N new training samples:

$$w_i \leftarrow \frac{1}{2} w_i, \tag{7.1}$$

if the ith classifier predicted the label wrong, otherwise w_i remains the same. The prediction of a new data instance is similar to MV, but using the adjusted w_i weights. Blum [20] also gives an upper bound for the M number of mistakes made by WMA:

$$M \leq 2.41(m + \log_{10} S), \tag{7.2}$$

where m is the number of mistakes made by the best expert and S is the number of experts.

A modified version of WMA is the Randomized Weighted Majority Algorithm (RWMA), also presented in [20]. In this algorithm

$$w_i \leftarrow \beta w_i \tag{7.3}$$

is applied when the ith expert predicts a label wrong (a good choice for β is proposed below). The upper bound for the M mistakes made by RWMA, dependent on the parameters m, S and β, is the following:

$$M \leq \frac{m \ln(1/\beta) + \ln S}{1 - \beta}, \tag{7.4}$$

proof can be found in [20]. Using this upper bound, Schapire [154] proposes to update β dynamically the following way:

$$\beta = \frac{1}{1 + \sqrt{\frac{2 \ln S}{m^*}}}, \tag{7.5}$$

where m^* is the number of mistakes made by the best classifier while the N labeled samples are processed sequentially. The modified w_i weights are used for the prediction of a new instance: the prediction of one selected classifier is used, where the ith classifier is selected with w_i/W probability, W being the sum of all weights. Although the upper bound given for RWMA is lower than for WMA, the practical use of this modification is questionable. RWMA suggests that, although the best expert of the ensemble is known, this expert should only be selected sometimes while other times one of the experts known to be worse should be relied on. Experiments presented later in Section 7.3.2 support this statement, showing that WMA generally performs better than RWMA.

Another approach presented here is the Weighted Majority Voting (WMV) [85]. In this algorithm a classifier's weight only depends on its p_i performance on the N labeled samples:

$$w_i = \log_{10}\Big(p_i/(1-p_i)\Big). \tag{7.6}$$

The prediction with WMV is similar to the MV and WMA methods.

In the experiments of Section 7.3.2 the above described methods MV, WMA, RWMA and WMV will be compared to each other and to the novel algorithm presented in the next subsection. Further methods exist based on the idea of weighted majority voting, but not fulfilling all the specifications given in Section 7.1.2. For example, Stefano et al. [164] present another version of weighting the majority vote rule. After training a set of experts for the general model, they define the search for the optimal w_i weights as a global problem (thus to maximize the performance of the whole set of experts), and apply a Genetic Algorithm (GA). They performed experiments on a handwritten digit recognition problem, showing that their proposed approach outperforms the traditionally used weighted majority voting approach where the weights are obtained based on only each single expert's performance. However, Stefano et al. [164] also state that their proposed approach requires a very high computational cost, and is thus not feasible for online mobile activity monitoring applications.

7.2.2 Dependent Experts

This section introduces a novel algorithm, called Dependent Experts (DE, *cf.* Algorithm 7.1). Similar to the various methods presented above, DE also uses a set of new labeled samples to train the weights of the experts, and uses weighted majority voting to predict a new, unlabeled data instance. The DE algorithm proposes a new approach to deal with the question of what is the confidence of an expert's decision when predicting a new, unlabeled sample. The main idea of DE is that this confidence should depend on the prediction of all other experts in the ensemble learner – thus the naming of this new algorithm. Therefore, the result of training the weights with the new labeled samples is a matrix of size SC (**W**, line 13 of Algorithm 7.1), where $w_{i,c}$ stands for the weight of the ith expert when the majority vote of all other experts is the class c (defined as the performance rate of the ith expert on this subset of samples, *cf.* line 8-10). In the prediction step, the label of the $\underline{x}_{new}$ instance is determined with an expert and with the ensemble of all other experts (*cf.* line 18 and line 19, respectively). The dependent weight obtained this way is added to the accumulated weight belonging to the label predicted by the respective expert (*cf.* line 20), repeating this procedure for each of the individual experts.

Existing weighted majority voting based methods only train an overall weight for each expert, while the $w_{i,c}$ weights make the DE algorithm a more flexible method: it supports the case when an expert is performing good on some classes, but poorly on others. As a consequence, DE also handles missing data better – *e.g.* when an expert has no knowledge on a part of the problem space. In the concrete case of personalization of activity recognition, s_i in Algorithm 7.1 refers to the classifier trained on the data of the ith subject, **C** is the set of separate activity classes, and **N** is the new labeled data from the previously unknown subject.

Algorithm 7.1 Dependent Experts

Require: **S** is the set of S different experts (classifiers): s_i, $i = 1, \ldots, S$
C is the set of C classes the classification task is composed of: c_i, $i = 1, \ldots, C$
N is the set of N new labeled samples: $\underline{n}_i = (\underline{x}_i, y_i)$, $i = 1, \ldots, N$ ($\underline{x}_i$: feature vector, $y_i \in [1, \ldots, C]$)
New instance to classify: $\underline{x}_{new}$

procedure TRAINING_WEIGHT(**S,C,N**)
  for $i \leftarrow 1, S$ **do**
    for $j \leftarrow 1, N$ **do**
      Predict label of $\underline{x}_j$ with expert s_i: $\hat{y}_j$
      Predict label of $\underline{x}_j$ with the ensemble $\mathbf{S} \cap s_i$ (all experts but s_i), using majority voting: $\hat{\hat{y}}_j$
    end for
    for $c \leftarrow 1, C$ **do**
      $\mathbf{P}_c = \{\forall \underline{n} \in \mathbf{N} \,|\hat{\hat{y}} = c\}$
        % *samples where the majority vote of the ensemble* $\mathbf{S} \cap s_i$ *is the class c*
      $\mathbf{P}_{c\_good} = \{\forall \underline{n} \in \mathbf{P}_c \,|\hat{y} = y\}$
        % *correctly predicted samples by* s_i *from the set of* $\boldsymbol{P}_c$
      $w_{i,c} = |\mathbf{P}_{c\_good}|/|\mathbf{P}_c|$
        % *the performance rate of the ith expert on* $\boldsymbol{P}_c$
    end for
  end for
  W is the return matrix of weights, composed of elements $w_{i,c}$ where $i = 1, \ldots, S$ and $c = 1, \ldots, C$
end procedure
procedure PREDICTION(**S,C,W,**$\underline{x}_{new}$)
  $\mu_c = 0$, $c = 1, \ldots, C$ % *initialize prediction of* $\underline{x}_{new}$
  for $i \leftarrow 1, S$ **do**
    Predict label of $\underline{x}_{new}$ with expert s_i: class $\hat{c}$
    Predict label of $\underline{x}_{new}$ with the ensemble $\mathbf{S} \cap s_i$: class $\hat{\hat{c}}$
    $\mu_{\hat{c}} \leftarrow \mu_{\hat{c}} + w_{i,\hat{\hat{c}}}$
  end for
  The output class is $\arg\max_c \mu_c \quad c = 1, \ldots, C$
end procedure

7.3 Experiments

This section first describes the basic conditions of the experiments, including the definition of activity recognition classification tasks and the decision on the used evaluation technique and performance measures. Afterwards, different aspects of the suggested personalization approaches are analyzed. In a thorough evaluation of the proposed general concept and the introduced novel DE algorithm, results are presented and discussed.

7.3.1 Basic Conditions

The basic conditions of the experiments are defined as follows. As in the previous chapters, the PAMAP2 dataset is used for the evaluation of the proposed general concept and new algorithm (*cf.* Section 3.3 for a more detailed description of the dataset). In order to analyze the proposed methods in different scenarios, a simple and a more complex physical activity recognition classification task is used in the experiments. First, the 'basic' activity recognition task (defined in Section 5.2.1) is used, which consists of 6 activity classes: lie, sit/stand, walk, run, cycle and Nordic walk. Moreover, as a complex classification problem, the PAMAP2_AR task is reused from Section 6.5. This task will be referred to as the 'extended' activity recognition task in this chapter. It consists of the following 15 activity classes: lie, sit, stand, walk, run, cycle, Nordic walk, iron, vacuum clean, ascend stairs, descend stairs, fold laundry, clean house, play soccer and rope jump.

The data processing chain defined in Section 4.2 is applied on the given raw data including preprocessing, segmentation and feature extraction steps. As a result 137 features are extracted, which serve as input for the classification step. Decision tree (DT) and AdaBoost.M1 (with DT as base-level classifier) are used and compared as classifiers in the general model. Decision trees were used in several previous works on personalization [120, 201], and are especially suitable for mobile applications, as pointed out in Section 5.2.2. Moreover, boosted decision tree classifiers were successfully applied on complex activity recognition tasks, shown *e.g.* in the benchmark results of Section 4.4.

The general model consists thus of several DT or boosted DT classifiers, all trained on single subjects from the dataset. These classifiers are all weighted the same in the original general model. The baseline performance of the experiments in this chapter is provided by MV, where no retraining of the weights is applied. The proposed new concept of personalization (using new labeled data to retrain only the weights of the individual classifiers) is evaluated with the existing methods WMA, RWMA and WMV. Moreover, these methods are compared to the newly introduced DE algorithm.

The leave-one-subject-out (LOSO) cross-validation method is used for evaluation to simulate the performance on new, unknown subjects. This means that data of one subject is left out from the training of the general model, then a certain amount of labeled data is used from the left-out subject to retrain the weights of the classifiers. The remaining data from the left-out subject is used for testing, repeating the entire procedure so that always another subject is left out.

Traditional performance measures are used to quantify the classification performance of the different methods: precision, recall, F-measure and accuracy. Moreover, since the range of individual accuracy of the subjects is of great importance, the highest and lowest individual subject accuracy is also used as measure. The focus is especially on the lowest individual accuracy: the goal of the proposed methods is to at least maintain the overall performance while significantly increasing the worst subject's performance.

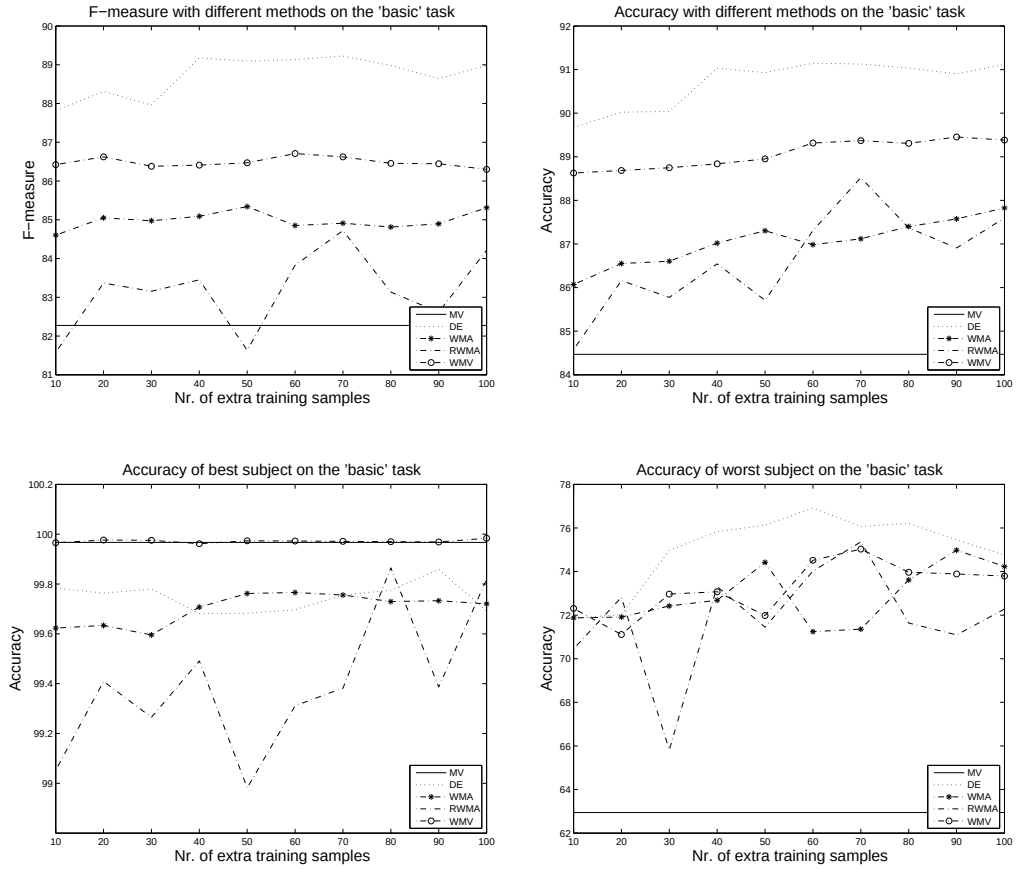


Figure 7.1: *Performance measures on the 'basic' task depending on the number of extra training samples per each activity class. Top row: the overall performance measures F-measure and accuracy. Bottom row: the highest and lowest individual subject accuracy.*

7.3.2 Results and Discussion

The experiments first specify a trade-off for the number of extra training samples required to retrain the weights in the general model. Moreover, a comparison of the different algorithms is given, both on the 'basic' and 'extended' classification task. Finally, the practical scenario is investigated when new labeled data from only a subset of activities is available.

Number of extra training samples

An important parameter of personalization approaches is the amount of data required to adapt the general model to a new subject. Figure 7.1 and Figure 7.2 show the performance measures of the different majority voting methods with DT classifier, while using 10-100 training samples per each activity class for the 'basic' and 'extended' task, respectively. In both figures, the lines belonging to MV show the case

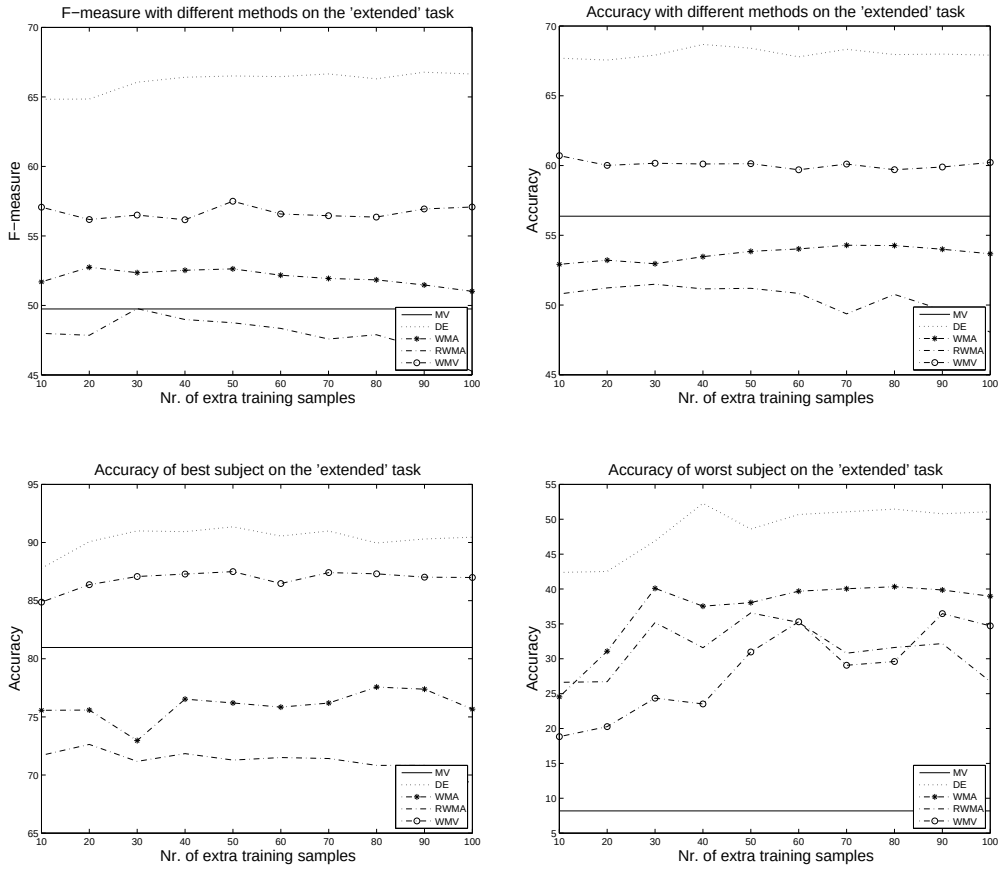


Figure 7.2: *Performance measures on the 'extended' task depending on the number of extra training samples per each activity class. Top row: the overall performance measures F-measure and accuracy. Bottom row: the highest and lowest individual subject accuracy.*

without retraining the general model. The results with the different weighted majority voting based algorithms suggest that the major increase in overall performance (*cf.* the plots of overall F-measure and accuracy) is achieved with just 10 extra training samples per activity. However, the lowest individual accuracy (*cf.* bottom right plot in both figures) significantly increases with more new training data, thus it is worth to select a higher number of extra training samples. On the other hand, the more new training samples are required the longer it takes to record labeled data and to retrain the weights for the new subject. Therefore, 60 extra training samples per activity are selected as trade-off, and used in the rest of this chapter. This means that the new subject has to record 1 minute of labeled data from each activity, since the applied data processing chain uses a sliding window of 1 second. The length of this additional data recording from the new user is comparable or even less than required in various previous works, as presented in Section 7.1.1.

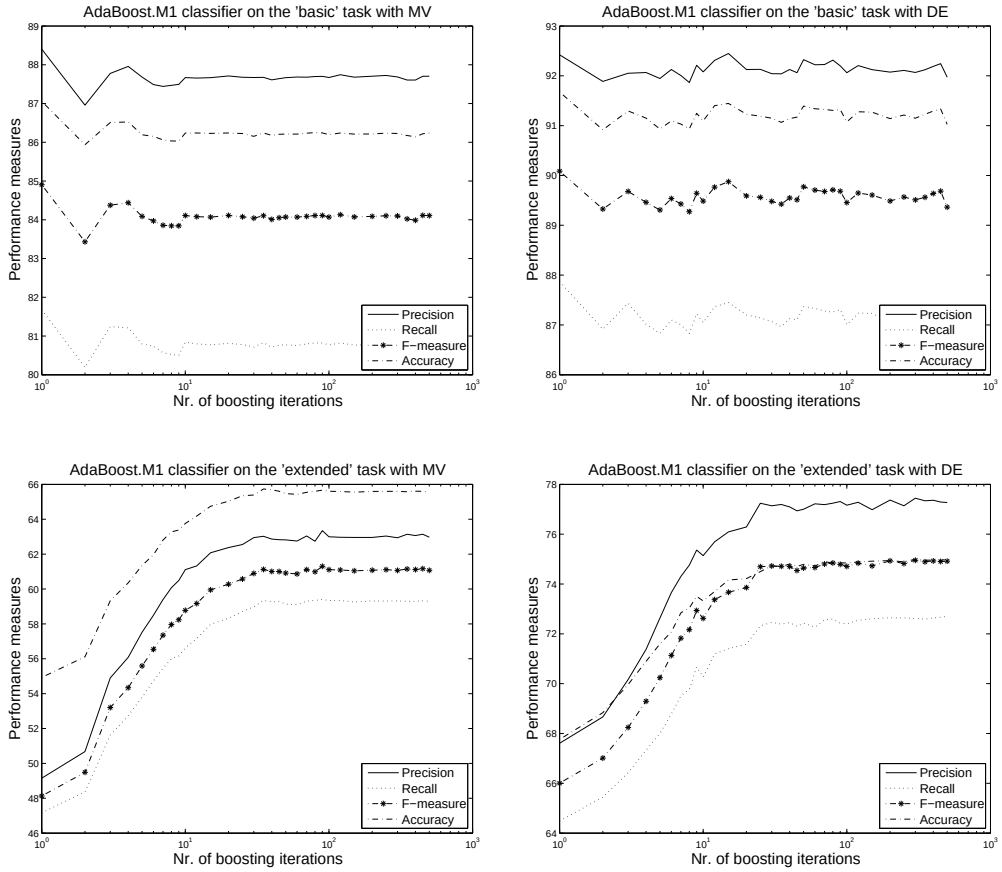


Figure 7.3: *Performance measures, depending on the number of boosting iterations, achieved with the AdaBoost.M1 classifier using the MV (left) and DE (right) algorithms. Top row: results achieved on the 'basic' task. Bottom row: results achieved on the 'extended' task.*

Comparison of the Algorithms

Table 7.1 and Table 7.2 present the results on the 'basic' task with decision tree and AdaBoost.M1 classifier, respectively. All results were achieved with 60 extra training samples per activity class, the selected trade-off for this parameter. Each of the experiments was performed 10 times, mean and standard deviation is given in the tables. Table 7.3 and Table 7.4 show the results on the 'extended' classification task with decision tree and AdaBoost.M1 classifier, respectively.

On the 'basic' task, an ensemble learner with DT classifier already performs well, no significant improvement can be achieved with the more advanced AdaBoost.M1 classifier (*cf.* the comparison of the respective results in Table 7.1 and Table 7.2). This statement is confirmed by the results shown in Figure 7.3 (top row, left and right plots presenting results with the MV and DE algorithm, respectively). On the other hand, the DT classifier only achieves rather low performance results on the 'extended' task. Therefore, it is worth to use a more complex classifier here, as shown in Figure 7.3

Table 7.1: *Performance measures on the 'basic' task with decision tree classifier and* 60 *extra training samples per activity class. The results are averaged over* 10 *test runs, mean and standard deviation is given for each experimental setup.*

	Precision	Recall	F-measure	Accuracy	Best subject	Worst subject
MV	86.46 ± 1.61	78.47 ± 1.58	82.27 ± 1.56	84.47 ± 1.44	99.97 ± 0.05	62.95 ± 6.01
WMV	90.25 ± 1.56	83.44 ± 1.53	86.71 ± 1.35	89.32 ± 1.19	99.97 ± 0.06	74.52 ± 3.11
WMA	86.86 ± 3.59	82.96 ± 3.60	84.85 ± 3.40	86.98 ± 2.70	99.77 ± 0.65	71.24 ± 5.63
RWMA	86.81 ± 3.49	81.08 ± 2.32	83.83 ± 2.55	87.31 ± 1.49	99.31 ± 0.75	74.02 ± 6.43
DE	91.63 ± 1.34	86.78 ± 1.43	89.13 ± 1.37	91.15 ± 0.97	99.70 ± 0.44	76.92 ± 4.87

Table 7.2: *Performance measures on the 'basic' task with AdaBoost.M1 classifier (100 boosting iterations) and* 60 *extra training samples per activity class. The results are averaged over* 10 *test runs, mean and standard deviation is given for each experimental setup.*

	Precision	Recall	F-measure	Accuracy	Best subject	Worst subject
MV	87.67 ± 1.45	80.77 ± 2.15	84.07 ± 1.73	86.20 ± 1.66	99.35 ± 1.51	66.25 ± 6.36
WMV	89.77 ± 5.05	86.71 ± 3.68	88.20 ± 4.27	89.91 ± 4.07	99.59 ± 0.44	71.65 ± 25.44
WMA	89.42 ± 3.52	85.32 ± 2.96	87.31 ± 3.01	89.33 ± 2.52	99.61 ± 0.35	77.56 ± 8.34
RWMA	87.11 ± 4.07	82.82 ± 4.76	84.87 ± 4.10	87.85 ± 2.98	99.01 ± 1.21	75.00 ± 8.08
DE	92.06 ± 1.46	87.00 ± 2.53	89.46 ± 1.98	91.07 ± 1.69	99.96 ± 0.07	76.91 ± 5.28

Table 7.3: *Performance measures on the 'extended' task with decision tree classifier and* 60 *extra training samples per activity class. The results are averaged over* 10 *test runs, mean and standard deviation is given for each experimental setup.*

	Precision	Recall	F-measure	Accuracy	Best subject	Worst subject
MV	51.11 ± 1.50	48.46 ± 1.16	49.74 ± 1.24	56.36 ± 1.48	80.97 ± 5.10	8.19 ± 3.66
WMV	58.88 ± 2.20	54.47 ± 1.38	56.58 ± 1.48	59.69 ± 1.46	86.47 ± 3.29	35.30 ± 9.02
WMA	52.93 ± 2.66	51.50 ± 2.66	52.17 ± 2.27	54.02 ± 1.41	75.84 ± 5.94	39.69 ± 6.02
RWMA	48.99 ± 2.61	47.78 ± 3.60	48.35 ± 2.97	50.83 ± 2.43	71.50 ± 3.76	35.21 ± 11.54
DE	68.21 ± 1.70	64.80 ± 1.55	66.46 ± 1.54	67.78 ± 1.59	90.56 ± 1.65	50.67 ± 5.20

Table 7.4: *Performance measures on the 'extended' task with AdaBoost.M1 classifier (100 boosting iterations) and* 60 *extra training samples per activity class. The results are averaged over* 10 *test runs, mean and standard deviation is given for each experimental setup.*

	Precision	Recall	F-measure	Accuracy	Best subject	Worst subject
MV	62.62 ± 1.06	59.61 ± 1.24	61.08 ± 1.07	65.32 ± 1.10	91.70 ± 1.64	18.55 ± 2.70
WMV	47.90 ± 3.12	55.00 ± 3.30	51.20 ± 3.18	54.58 ± 3.37	96.79 ± 0.54	15.22 ± 9.82
WMA	73.11 ± 1.99	67.19 ± 2.35	70.02 ± 2.10	68.13 ± 1.56	89.95 ± 4.01	53.62 ± 2.84
RWMA	69.36 ± 4.23	64.37 ± 2.35	66.74 ± 2.83	65.71 ± 1.43	91.44 ± 2.65	54.14 ± 8.87
DE	76.83 ± 1.09	72.14 ± 1.22	74.41 ± 1.14	74.79 ± 1.13	96.03 ± 1.25	56.14 ± 2.80

(bottom row, left and right plots presenting again the results with the MV and DE algorithm, respectively): increasing the number of boosting iterations significantly improves on the different performance measures. It should be noted that the results of AdaBoost.M1 in Table 7.2 and Table 7.4 are all given with 100 boosting rounds, as the performance already levels off at this iteration number.

From the presented results it is clear that the concept of retraining the weights of a general model is a valid approach for the personalization of physical activity recognition. Compared to the baseline performance of MV, the overall performance measures of all weighted majority voting algorithms are at least comparable, while the lowest individual performance increases significantly. This is true with the methods WMA and RWMA, but overall the WMV algorithm performs best amongst the existing approaches. Moreover, the novel DE algorithm clearly outperforms all existing methods, both in overall performance and in increasing the worst subject's performance. Therefore, this new method is a very promising approach for personalization.

Number of retraining activities

In practical scenarios it is not always feasible that a new user of an activity recognition application records labeled data for each activity class to personalize the general model. For example, concerning the 'extended' task, activities such as *rope jump* or *vacuum clean* might not be of interest for each of the new users, or they could lack the necessary equipment to perform them. Therefore, this section also investigates the behaviour of the proposed concept and new algorithm when new labeled data only from a subset of the activity classes is available.

Figure 7.4 and Figure 7.5 show results with DT classifier on the 'basic' and 'extended' task, respectively. The number of activities from which new data is available to retrain the weights is changed from 0 (no retraining, equivalent to MV) to 6/15 (thus new data from each activity is provided, equivalent to the results given in Table 7.1 and Table 7.3, respectively). Each experiment is performed 10 times, selecting random the activity classes for retraining. The results show that even when fewer activities are performed by the new subject the performance increases compared to MV, especially concerning the lowest individual accuracy. Similar to the above presented results, the existing method WMV and the novel DE algorithm perform best in these experiments. For example, if the new subject only records data from half of the recognized activity classes (thus 3 activities in case of the 'basic' task and 7 – 8 activities in case of the 'extended' task), while the overall performance does not decrease, the lower boundary of the subject's individual performance already significantly increases. This means that the overall similarity of the new subject to each of the training subjects can be learnt to some extent, even when only limited new training data is available.

7.4 Computational Complexity

The main goals defined in Section 7.1 were that the novel personalization concept supports more advanced classifiers while still feasible for mobile applications regard-

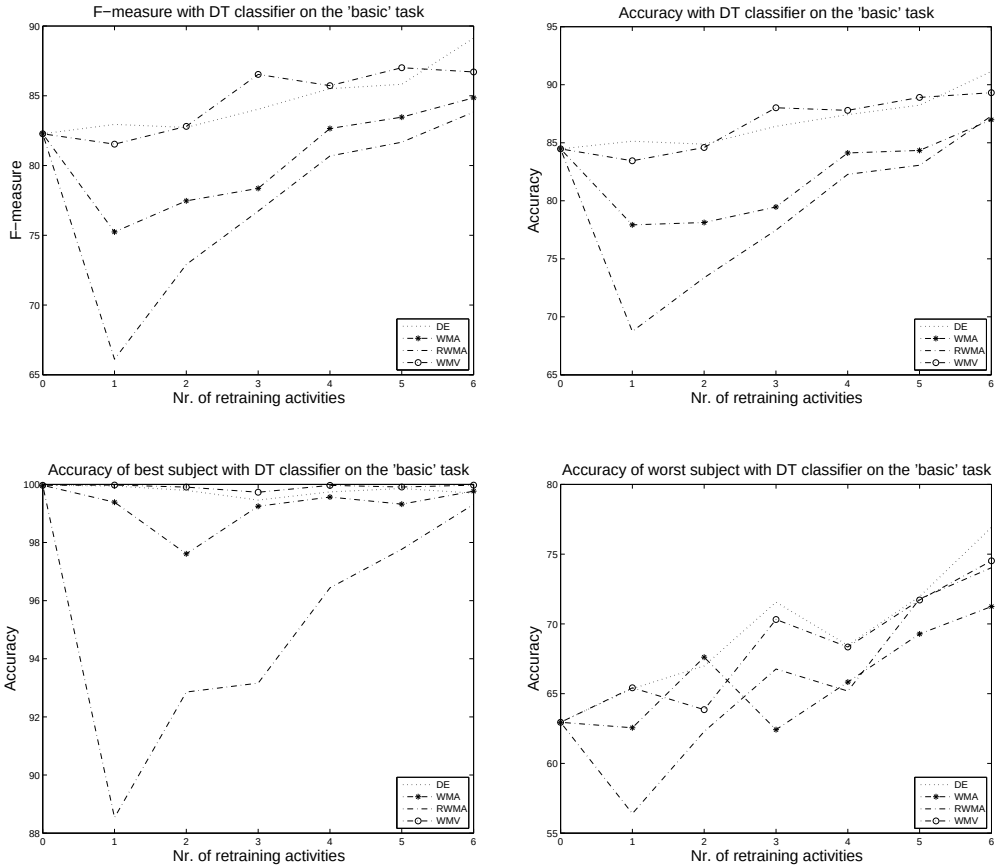


Figure 7.4: *Performance measures on the 'basic' task depending on the number of activities (0 – 6) from which new data is available to retrain the general model. Top row: the overall performance measures F-measure and accuracy. Bottom row: the highest and lowest individual subject accuracy.*

ing its computational complexity. The results presented in Section 7.3 show that the personalization of complex classifiers (AdaBoost.M1 as an example) is possible with both the new concept and the novel DE algorithm. Therefore, this section will analyze the computational complexity of the proposed methods on mobile systems.

An empirical study is designed and carried out to investigate the feasibility of the new personalization approach for mobile physical activity recognition. This study is performed using the mobile system described in Section 8.3.1, thus using wearable wireless sensors and a Samsung Galaxy S III smartphone (this device contains a 1.4 GHz quad-core Cortex-A9 CPU and 1 GB of RAM). The procedure of the empirical study can be described as follows. First, data is recorded from one subject during 6 sessions, each session including the following 7 activities: lying, sitting, standing, walking, running, ascending and descending stairs. These recordings were used to create the general model, consisting of 6 classifiers (each of these classifiers was trained using data from one of the sessions). Decision tree, AdaBoost.M1 and Conf-

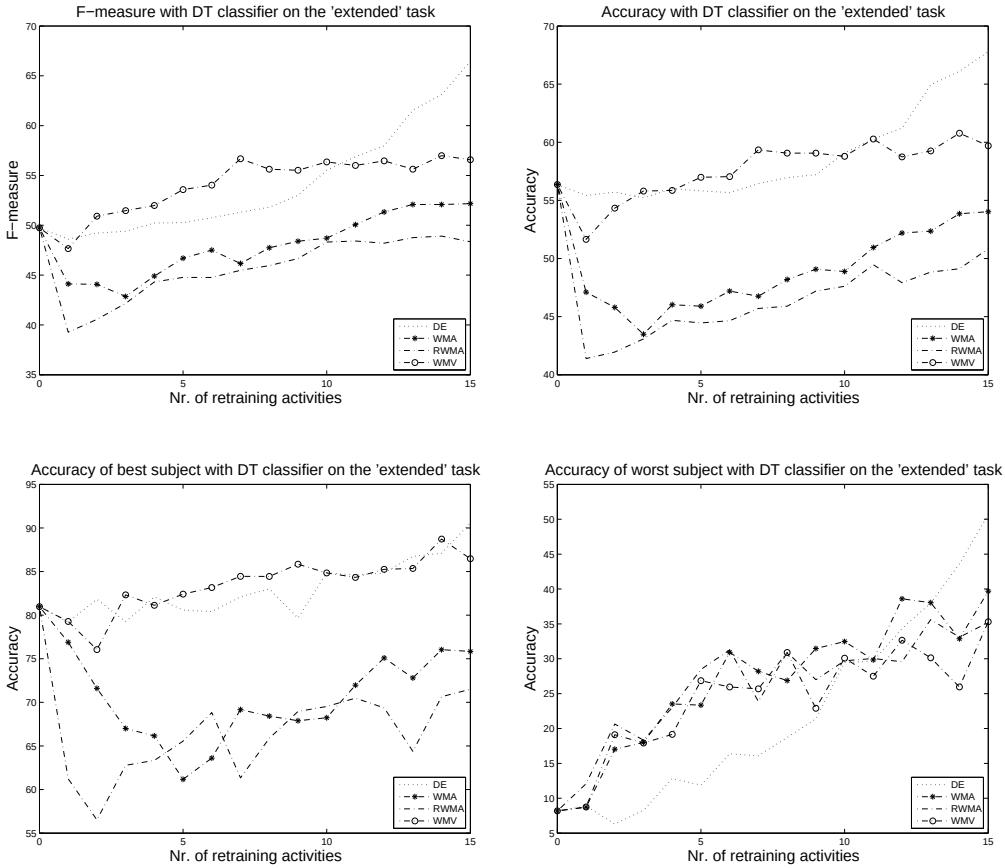


Figure 7.5: *Performance measures on the 'extended' task depending on the number of activities (*$0 - 15$*) from which new data is available to retrain the general model. Top row: the overall performance measures F-measure and accuracy. Bottom row: the highest and lowest individual subject accuracy.*

AdaBoost.M1 (both with decision tree as base-level classifier) were used and compared as classifiers in the general model. Then, labeled data from a second subject is recorded, while performing each of the 7 activities for approximately one minute (as defined in Section 7.3.2). This new data is used to retrain the weights of the 6 classifiers of the general model. The retraining of the weights is performed directly on the smartphone, for each of the 4 analyzed algorithms. For each classifier – personalization algorithm pair the retraining was run 5 times, results will present the average of these test runs. Finally, to compare the performance of the different methods using the mobile system, the second subject also recorded data for offline evaluation, performing each of the 7 activities for approximately three minutes. However, it should be noted that the main goal of this empirical study was not to compare classification accuracy of the methods, but to analyze and compare the computational time of the proposed algorithms.

Table 7.5: *Computational time [s] required for the retraining of the general model. Decision tree, AdaBoost.M1 and ConfAdaBoost.M1 are each tested as classifiers used in the general model. The proposed personalization approach is evaluated with the weighted majority voting-based methods WMV, WMA and RWMA, and the novel DE algorithm.*

Classifier	WMV	WMA	RWMA	DE
Decision tree	4.01	3.89	4.01	4.05
AdaBoost.M1	10.91	11.03	10.82	10.84
ConfAdaBoost.M1	30.89	30.48	31.01	31.00

Table 7.6: *Classification accuracy [%] of each of the majority voting-based algorithms and each type of classifier applied in the general model.*

Classifier	MV	WMV	WMA	RWMA	DE
Decision tree	84.64	87.61	92.44	92.44	88.54
AdaBoost.M1	80.14	86.08	84.38	84.38	85.74
ConfAdaBoost.M1	84.97	92.70	92.28	92.28	92.19

Table 7.5 presents the average retraining time of each of the weighted majority voting algorithms and each type of classifier. The interpretation of these results is the following: After the second subject recorded the required new training data and started the retraining of the general model on the smartphone, how long did he have to wait to receive his personalized model. For each of the retraining algorithms, the major computational time is spent to predict the label of each new sample by each of the general model's classifier (for which the by far most computationally intensive part is the feature calculation in the DPC). With an effective implementation this has to be done exactly once for each sample – classifier pair, even when applying the DE algorithm. Therefore, the retraining time for all 4 algorithms should be similar, as proved by the results of Table 7.5. Furthermore, the retraining of the general model when consisting of ConfAdaBoost.M1 classifiers takes the longest, since this classifier is the most complex, thus includes the calculation of the most features. Nevertheless, the retraining time of approximately 30 seconds is still acceptable: The new user receives a complex personalized system after only waiting half a minute, which is far below the required time presented in related work.

Table 7.6 shows the classification accuracy for each of the majority voting algorithms and each type of classifier. These results were achieved with the different personalized models, on the data recorded by the second subject for offline evaluation purposes (approximately three minutes for each of the 7 activities). Since the amount of data used for this evaluation is rather small, no statistically significant conclusion can be drawn. Nevertheless, these results serve as proof of concept, showing that the novel personalization concept is realized and successfully trained on the proposed mobile system. The accuracy of the general model's single classifiers in case of using decision tree classifier ranges between 45.08% and 92.44%, in case of AdaBoost.M1 it ranges between 45.08% and 90.49%, and in case of ConfAdaBoost.M1 it ranges

between 45.08% and 92.28%. Therefore, no significant difference can be observed between the three types of classifiers. A larger number of subjects and a wider range of activities would be required for evaluation, as shown in Chapter 6. The results of Table 7.6 show that the retraining of the weights improves on the classification results for each of the algorithms, compared to MV. However, a more complex classification task would be required to observe the DE algorithm outperforming the other methods, as presented in Section 7.3.

7.5 Conclusion

This chapter presented a novel general concept for the personalization of physical activity recognition applications. This concept uses a set of classifiers as general model, and retrains the weight of the classifiers using new labeled data from a previously unknown subject. Results with different methods based on this concept (using WMA, RWMA and WMV algorithms) show that it is a valid approach. Moreover, a novel algorithm is presented and compared to the existing methods, further increasing the performance of the personalized system. These statements are confirmed with a thorough evaluation on two activity recognition classification tasks, comparing also decision tree and boosted decision tree classifiers as experts used in the general model.

The main benefit of the introduced concept is that, instead of retraining classifier(s) of a general model, only their weights are retrained. This is much less computationally intensive, since basically only the prediction of the new training samples is required. Therefore, the proposed approach can also be used for mobile systems, even for complex classification tasks requiring more complex classifiers (*cf.* the 'extended' task with AdaBoost.M1 classifier). An analysis of the computational complexity of the new personalization concept shows its feasibility for online mobile applications. A new user receives the personalized model within a short time. Moreover, the proposed concept allows that the new user only records data from a subset of the recognized activities, making the approach more practicable.

Physical activity monitoring systems are usually trained on a user group of young, healthy adults. Without applying personalization to such applications, they only perform poorly when used by significantly differing users, *e.g.* overweight or elderly subjects. In future work it is planned to investigate how well personalized systems perform in these situations, how much improvement the proposed personalization concept and novel DE algorithm achieves compared to when only applying a general model.

8

Physical Activity Monitoring Systems

8.1 Introduction

The previous chapters presented novel algorithms in order to improve the classification performance of physical activity monitoring systems. This chapter describes how an actual activity monitoring system can be created, various issues concerning such systems are discussed.

The main goal of this chapter is to create a mobile, unobtrusive activity monitoring system. For collecting inertial and physiological data small, lightweight and wireless sensor units should be used, as described in Section 8.3.1. Moreover, as few sensor positions as possible should be utilized on the user's body. A thorough analysis concerning this issue for both intensity estimation and activity recognition is given in Section 8.2. For processing the collected data current smartphones are chosen for the final prototype, as discussed in Section 8.3.1. Two feasibility studies are carried out in Section 8.3.2 to show that the computational power of such mobile devices is sufficient for applying complex classification algorithms. Moreover, the mobile device is also used for providing the user with feedback , visualizing the results as described in Section 8.3.3.

A further goal of this chapter is the integration of the described mobile system into a full healthcare application for aerobic activity monitoring and support in daily life. The major components of such an overall system and how they interact with each other are presented in Section 8.4. Finally, the chapter is summarized in Section 8.5.

8.2 Modular Activity Monitoring System

This section analyzes how many and which sensors are required for a reliable physical activity monitoring system. The importance of different sensor placements is investigated for intensity estimation and activity recognition. A similar study was carried out for only activity recognition by Dalton and OLaighin [38]. They reached an accu-

Table 8.1: *Modular activity monitoring system: results on the intensity estimation task with different combinations of the sensors.*

chest IMU	arm IMU	foot IMU	heart rate	Performance [%]
X				90.47
	X			86.47
		X		88.08
			X	82.06
X			X	94.37
	X		X	93.07
		X	X	91.36
X	X	X		94.07
X	X	X	X	95.65

racy of 95% with LOSO protocol using 5 sensor positions, while with only 2 sensors (wrist and ankle location) the accuracy still reached 88%.

In this thesis both intensity estimation and activity recognition are considered. Results achieved on these tasks with different combinations of sensors are presented in Section 8.2.1 and Section 8.2.2, respectively. Moreover, the results motivate to introduce the concept of a modular activity monitoring system. As a concrete example a system consisting of three modules is described here. By using different combinations of these modules, different functionality becomes available: 1) a coarse intensity estimation of physical activities 2) different features based on heart rate data and 3) the recognition of basic activities and postures.

All below presented results are based on the PAMAP dataset, which provides data from three IMUs (located at a subject's arm, chest and foot) and a heart rate monitor, *cf.* Section 3.2. The data processing chain as defined in Section 4.2 is used. Boosted decision tree classifier is applied in this chapter since it performed best in the preliminary studies of Section 4.2.4. Furthermore, LOSO 8-fold cross-validation protocol is applied. All experiments are performed using the Weka toolkit [65].

8.2.1 Intensity Estimation

The intensity estimation task introduced in the preliminary studies of Section 4.2.4 is used in this subsection. This task defines 3 classes: The goal is to distinguish activities of light, moderate and vigorous effort. Each of the 14 activities included in the PAMAP dataset is assigned to one of these intensity classes. This coarse intensity estimation is sufficient in many applications, *e.g.* to monitor how individuals meet health recommendations [66].

Table 8.1 shows results on the intensity estimation task with various sets – combinations which are considered to be of interest for this task – of sensors. One row in the table represents one setup, crosses in the four columns indicate which sensors are included in a specific setup. The results show that from the three IMU positions investigated the chest placement performs best. Moreover, by adding the HR-monitor very good (94.37%) results are achieved for this task. By adding two more accelerometers

Table 8.2: *Modular activity monitoring system: results on the activity recognition task with different combinations of the sensors.*

chest IMU	arm IMU	foot IMU	heart rate	Performance [%]
X				83.36
	X			73.55
		X		74.67
			X	45.64
X			X	83.85
	X		X	77.55
		X	X	76.45
X	X		X	88.11
X		X	X	81.70
	X	X	X	89.95
X	X	X		88.90
X	X	X	X	90.65

(on arm and foot placement) further improvement on intensity estimation is obtained. However, it is questionable whether it is worth using two extra sensors for only a minor improvement in performance. On the other hand, if the two extra accelerometers are required for other tasks in an activity monitoring system (*e.g.* for activity recognition), features derived from them are used for intensity estimation as well. Moreover, features combining different sensor locations (*e.g.* the weighted sum of the absolute integral of all three accelerometers, *cf.* Section 4.2.3) are applied as well. Therefore, if synchronized data from different sensor placements is available, it is worth extracting and investigating features computed from multiple sensor locations for the intensity estimation task.

The results in Table 8.1 also indicate that – in contrast to the conclusion of [173] – heart rate information combined with accelerometers improves the intensity estimation of physical activities, compared to systems only relying on inertial data. This is especially true for walking-like activities of light/vigorous effort. Without using the HR-monitor, the performance of the intensity estimation is poor on the activities *very slow walk* and *ascend stairs*. The reason is that the characteristics of these activities overlap with *normal walk* if only considering features extracted from accelerometer data. This justifies the need of features extracted from physiological measurements, *e.g.* from heart rate data. However, the results of Table 8.1 also show that heart rate information alone is not sufficient for a reliable intensity estimation.

8.2.2 Activity Recognition

Table 8.2 shows results on the activity recognition task with various sets of sensors. Compared to Table 8.1 it is clear that the activity recognition task defines a more difficult classification problem, than the intensity estimation task does. When using only the chest IMU and the HR-monitor – the most efficient setup for the intensity estimation task – only a relatively low performance (83.85%) can be achieved. Therefore,

the usage of the two extra accelerometers is justified: The overall performance can be increased to 90.65%.

An interesting conclusion from the results of Table 8.2 (from the performance results on the setups containing two IMUs and the HR-monitor) is that the chest and foot IMU placements behave similarly for activity recognition, while the arm IMU placement is complementary. Comparing the activity type of misclassified samples with and without using the arm IMU reveals that distinguishing *normal walk* and *Nordic walk* is effectively not possible without using the arm IMU.

8.2.3 Conclusion

With recent progress in wearable sensing the number of commercially available activity monitoring products is increasing. Most of these products include one sensor, located on the user's body (*e.g.* as a bracelet, on the belt or directly integrated in a mobile device), and focus on a few goals usually related to the assessment of energy expenditure. Studies underline the good accuracy of some of these systems, *e.g.* the Actiheart [36] or the SenseWear [82] system. However, there exist different needs towards an activity monitoring system. Additional functionality is introduced in some of the above mentioned products, *e.g.* the assessment of sleep duration and efficiency in the SenseWear system. However, there is no possibility to extend these systems if *e.g.* a higher accuracy or more information is required for adding further functionality related to physical activity monitoring.

The results presented in Section 8.2.1 and Section 8.2.2 indicate that a different set of sensors is required for different physical activity monitoring tasks. This motivates the idea of introducing a modular activity monitoring system: By adding or removing sensors different functionality can be added or removed. The rest of this section describes an extensible physical activity monitoring system based on this idea: Given a simple system for the intensity estimation of physical activities, a more detailed description of daily activities can be acquired with one or two additional set of sensors.

The basic system consists of only one accelerometer worn on the chest. This delivers a reliable coarse intensity estimation of physical activities, *cf.* Table 8.1. By adding a heart rate monitor, the following benefits can be achieved compared to the basic system: 1) a significantly improved intensity estimation and 2) new functionality is available based on the obtained heart rate information. The first benefit is justified by the results of Table 8.1, since the performance increased by approximately 4% with the additional heart rate monitor (from 90.47% to 94.37%). As for the second benefit: Monitored heart rate can extend the functionality of an activity monitoring system in many ways. For cardiac patients for example, a specific HR could be defined individually in the system, and an alarm would be initiated when exceeding this value. For sports applications, a desired range of HR can be defined, and the system can determine how much time was spent in this heart rate zone to optimize the benefits from a workout.

Finally, by adding two extra accelerometers (arm and foot placement) to the basic or the HR-monitor extended system – besides a further improvement on the intensity

estimation – the recognition of basic activities and postures is enabled. This module is justified by the results shown in Table 8.2: an accuracy of 88.90% or 90.65% was achieved on an activity recognition task with the 3 IMUs or the 3 IMUs and the heart rate monitor, respectively.

As a result, the idea of a modular system for physical activity monitoring was presented within this section: a base module is responsible for the basic system functionality (intensity estimation in the concrete example of this section), while two more modules can be added – separately or together – to extend the functionality of the system. Following the idea of modularity, additional modules could be defined. A possible extension to the presented system is *e.g.* a module providing full upper-body tracking. Therefore, besides the already provided monitoring of aerobic activities, the monitoring of muscle-strengthening activities would become available.

8.3 Mobile Activity Monitoring Systems

The first mobile prototype developed within the PAMAP project [116] was presented in Section 3.2.1. This early prototype was used to record the PAMAP physical activity monitoring dataset. It included 3 wired Colibri inertial measurement units from Trivisio, the Garmin Forerunner 305 GPS-enabled sports watch with integrated heart rate monitor, and a Sony Vaio UMPC as collection unit. This system had several major drawbacks, *e.g.* the very limited battery time of the collection unit and the wiring to connect the sensors. Therefore, a practical usage of this prototype in everyday life was not feasible.

The PAMAP2 dataset was recorded with an improved prototype of the system, as described in Section 3.3.1. It included 3 Colibri wireless IMUs from Trivisio, a Bluetooth-based heart rate monitor from BM innovations GmbH, and the Viliv S5 UMPC as collection unit. All subjects participating in the data capturing reported that the sensor fixations were comfortable and did not restrict normal movements. A drawback of this system was the custom bag required for the collection unit and the additional USB-dongles for wireless data transfer.

The PAMAP2 prototype was also tested in a clinical trial within the PAMAP project, *cf.* [116]. 30 elderly subjects participated in this study, including both healthy elderly, and cardiovascular and functional disease patients. Each of the subjects was instructed first about the system, then they kept it over one week to monitor their daily activities. A positive observation during these trials was that attaching the sensors and other hardware components was straightforward, it could be done alone by the elderly subjects. The entire setup took not longer than 5 minutes. Moreover, the system's battery time of approximately 6 hours was sufficient to cover the active part of a subject's normal day. A few subjects complained about the somewhat bulky sensors, especially on the arm and chest placements. The main concern of most of the elderly was related to the collection unit and the custom bag, which felt sometimes uncomfortable during intensive movements. Furthermore, some of the subjects expressed their dislike of the custom bag from the esthetic point of view, thus they would not wear it during their daily routine.

Overall, although the PAMAP2 prototype was in general feasible for data collection, some drawbacks still limit the system's usability in everyday life. Therefore, further improvement is required, concerning especially the collection unit and the size of the sensor units. The next subsection presents a state-of-the-art prototype for mobile physical activity monitoring. It is based on commercially available, widely used sensors and an Android smartphone, making it more acceptable for everyday usage.

8.3.1 Final Prototype

The final prototype consists of Shimmer wearable wireless sensors, a wireless heart rate monitor and an Android smartphone. Shimmer is a small, low power wireless platform [160], widely used in wearable sensing research and by clinical, rehabilitation and care delivery professionals. The platform consists of a baseboard, which can be extended by different sensor modules (*e.g.* 9 DoF IMU sensor, GPS, GSR, ECG or EMG module). This modular setup makes the Shimmer platform flexible and configurable. The baseboard includes an MSP430 microcontroller, a Roving Networks RN-42 Bluetooth module [114] and an integrated 3-axis accelerometer. It further includes a 802.15.4 radio module and supports on-device data storage with an integrated microSD card slot. Shimmer's firmware is developed in the nesC programming language, thus it works with the TinyOS operating system. TinyOS is designed for low-power wireless devices, such as those used in sensor networks, ubiquitous computing and personal area networks [176]. Furthermore, Shimmer provides an Android instrument driver, allowing to stream data directly to Android devices. This driver is especially useful for research purposes, since it significantly reduces the application development time.

As pointed out in Section 4.1, accelerometers are by far the most useful inertial sensors for physical activity monitoring. The analysis in Section 8.2 showed that 3 sensor placements are required for a reliable activity recognition. Therefore, the proposed final prototype includes 3 Shimmer baseboard units (since this already provides an accelerometer, no extension modules are needed for kinematic sensing). These units are lightweight (22 g with battery and enclosure, compared to 48 g of the Colibri wireless units from Trivisio) and small ($53 \times 32 \times 15$ mm). The integrated Freescale MMA7361 accelerometer has a configurable range of either ± 1.5 g or ± 6 g, with a sensitivity of 800 mV/g at ± 1.5 g. The sampling rate of the sensor is also configurable, enabling maximum 400 Hz for the x and y axes, and maximum 300 Hz for the z-axis. The Shimmer sensor units are placed on chest, lower arm and ankle positions, since this sensor setup proved to be beneficial in the PAMAP2 data collection, *cf.* Section 3.3.1. The sensors can be fixated with the available wearable straps from Shimmer, ensuring easy setup and comfortable wearing of the sensors.

Shimmer has announced the release of a heart rate monitor extension, but this module is not available yet (this chapter is written in August 2013). As an alternative, the Zephyr Bioharness 3 wireless heart rate monitor is included in the proposed final prototype [199]. This sensor uses also Bluetooth to stream heart rate data, providing a range of 25 – 240 BPM (beats per minute).

As for mobile control unit, a general Android smartphone is proposed for the final prototype. The main tasks of the control unit are data processing and visualization of the results. The choice of a smartphone for these tasks is preferable, since this way no additional device is required as control unit, most users would anyhow carry a smartphone with themselves during their daily routine. Moreover, these devices support wireless data transfer by Bluetooth, thus the selected sensors (Shimmer units and the Zephyr heart rate monitor) can directly stream the collected data to the control unit for online processing, no additional hardware component (*e.g.* dongle) is required. The Android operating system was selected for its comfortable way of developing applications, good support for external devices (*cf. e.g.* the Android instrument driver from Shimmer) and its large community of developers. Two Android smartphones were tested within the final prototype, namely the Google Nexus S (available since December 2010, including a 1GHz single-core ARM Cortex-A8 CPU and 512MB of RAM) and the Samsung Galaxy S III (available since May 2012, including a 1.4GHz quad-core Cortex-A9 CPU and 1GB of RAM).

The entire data processing chain (*cf.* Section 4.2) is implemented in `Java` for the Android smartphones, resulting in an online application for long-term physical activity monitoring. The feature extraction step is optimised in a way that each feature should be computed at most once on each window segment. As for the classification step, the ConfAdaBoost.M1 algorithm with C4.5 decision tree as weak learner is chosen for the implementation, since this is the best performing classifier throughout this thesis (*cf.* Chapter 6). Feasibility studies of applying such complex classifiers on mobile devices are carried out in Section 8.3.2.

Both intensity estimation and activity recognition are included in the mobile application. For both tasks the definition as presented in Section 4.4.1 is used (the background task for activity recognition), thus 3 intensity and 7 activity classes are to be distinguished. For dealing with the other activities in the activity recognition task the 'bgClass' model is used, as this has the best generalization characteristics (*cf.* Chapter 5). The boosted decision tree classifiers for both the intensity estimation and activity recognition tasks were trained using the PAMAP2 dataset.

Apart from the implemented data processing chain, the mobile application of the final prototype also provides a graphical user interface (GUI) for the user. This includes on the one hand a labeling tool similar to the one presented in Section 3.3.1. Therefore, further data collection can be performed with the proposed final prototype, offering a robust and unobtrusive system for this purpose. On the other hand, the GUI also provides feedback to the user, visualizing the results on the smartphone's display. The type of feedback is described in detail in Section 8.3.3, providing also visualization examples of the mobile application.

8.3.2 Using Complex Classifiers: Feasibility Studies

The best performing classifiers throughout this thesis were the different boosted decision tree classifiers. They have several further benefits, *e.g.* they possess a simple structure (it can be basically described as a large if-then-else structure), and are thus easy to implement. However, boosting is complex in the way that the size of the clas-

Table 8.3: *Feasibility study I: comparing size and average computational cost of classification of the three different decision tree (DT) classifiers on a mobile device (Viliv S5 UMPC).*

Classifier	Size	No. of leaves	Computation time (ms)	No. of features
Custom DT	15	8	4.9	3.57
C4.5 DT	119	60	23.6	8.82
Boosted DT	1464	737	184.3	79.78

sifier is about T-times larger (T being the number of boosting iterations) than the applied base-level classifier. This means that boosted classifiers have larger computational requirements. Therefore, the question arises whether such complex classifiers are feasible for online activity monitoring applications: Due to the mobile systems these applications are running a restriction on available computational power exist. This subsection describes two empirical studies performed to examine this question.

Study I: AdaBoost.M1

The first feasibility study compares AdaBoost.M1 (with C4.5 decision tree as weak classifier) to two other decision tree classifiers: a custom decision tree classifier (*cf.* Figure 4.4) and a C4.5 decision tree classifier. The two latter classifiers were selected since each represents a different complexity level and all three classifiers have a binary tree structure, thus a comparison of them is straightforward. All three classifiers were introduced in Section 4.2.4.

All three classifiers to be compared were implemented in `C++` on a Viliv S5 UMPC (the control unit used for the PAMAP2 data collection, *cf.* Section 3.3.1), containing an Intel Atom Z520 CPU (1.33GHz) and 1GB of RAM. The structure of the implementation includes a data collection thread (including preprocessing and segmentation of raw sensory data) for each of the sensors, and a data processing thread for feature extraction and classification. The training of each of the three classifiers was done offline using the PAMAP dataset. Then, the trained binary tree structures were converted into `C++` code for online classification on the UMPC. With each of the classifiers an approximately 15 minutes protocol was followed wearing the mobile system. This protocol included a wide range of activities (lying, sitting, standing, walking, running, ascending stairs and descending stairs) to be able to observe the classifiers in different states of their function.

Table 8.3 shows the comparison of size and average computational cost of the three different decision tree classifiers. The size of the classifiers in the table refers to the number of decision and leaf nodes together. The computation time includes the classification and the computation of the required features for the respective classification step, and was computed in the above mentioned data processing thread of the online application. The number of features in the table refers to the average number of computed features per window segment. It should be noted that the above mentioned optimization of the feature extraction step (computing each feature at most once on each window segment) is applied here.

Table 8.4: *Feasibility study II: comparing computational cost and performance of the three different C4.5 decision tree (DT) based classifiers on a smartphone (Samsung Galaxy S III).*

	Computation time (ms)		Accuracy [%]	
Classifier	**average**	**maximum**	**intensity est.**	**activity rec.**
C4.5 DT	3.32	42	94.36	93.84
AdaBoost.M1	24.54	131	96.18	98.98
ConfAdaBoost.M1	51.54	150	99.79	100

Although the results in Table 8.3 show that the computational cost of the boosted decision tree classifier is an entire order of magnitude higher than the computational cost of the C4.5 decision tree classifier, it is still far below the restriction given by the application. This restriction is defined by the fact that the segmentation step of the DPC uses a sliding window shifted by 1 second, thus the data processing thread has maximum 1 second for each processing step. Therefore, this empirical study showed that the more complex boosted decision tree classifier is a considerable choice even for mobile activity monitoring applications, there are no limitations considering the computational costs.

Study II: ConfAdaBoost.M1

The second empirical study is carried out with the final prototype of the mobile system, described in Section 8.3.1. The main goal of this study is to show the feasibility of the ConfAdaBoost.M1 algorithm on mobile devices. ConfAdaBoost.M1 is compared to a C4.5 decision tree classifier and to AdaBoost.M1. Both boosting classifiers have the C4.5 decision tree as weak learner.

All three classifiers to be compared are implemented in `Java` on a Samsung Galaxy S III smartphone, which contains a 1.4GHz quad-core Cortex-A9 CPU and 1GB of RAM. The structure of the implementation is similar to the one of the above described first empirical study. First, data is recorded from two subjects performing various activities while wearing the mobile system. This data is used for training all classifiers, for both the intensity estimation and activity recognition tasks. Then, with each of the trained classifier an approximately 20 minutes protocol is carried out by one of the subjects. The same protocol is followed with each classifier, including the following wide range of activities: lying, sitting, standing, walking, running, cycling, ascending and descending stairs.

Table 8.4 shows the comparison of the three different decision tree-based classifiers. It is clear that – similar to the previous study – even the maximum computation time of each classifier is far below the restriction of 1 second. The difference between AdaBoost.M1 and ConfAdaBoost.M1 in computation time can be explained by the fact that the training of the AdaBoost.M1 algorithm stops at an earlier boosting round (as discussed in Chapter 6), thus this classifier is of smaller size. With the ConfAdaBoost.M1 algorithm on the other hand, the predefined iteration number of 100 is reached during the training for both intensity estimation and activity recognition.

Apart from proving the feasibility of using ConfAdaBoost.M1 for online activity monitoring on smartphones, this second study serves also as proof of concept: The proposed final prototype of the mobile system was fully realized and tested. Both the provided labeling tool and the implemented data processing chain were working well, thus overall a fully functional, robust and unobtrusive mobile physical activity monitoring system was realized in this section. Moreover, the comparison of the three classifier's performance in Table 8.4 confirms previous results of this thesis: The ConfAdaBoost.M1 algorithm outperforms the other classifiers on both the intensity estimation and the activity recognition tasks. Although these results were achieved with data from only two subjects and the results are subject dependent, they show a clear tendency and thus justify applying ConfAdaBoost.M1 in the implemented data processing chain.

8.3.3 Feedback, Visualization

The previous chapters of this thesis presented various methods for physical activity monitoring, while the previous sections of this chapter described the creation of a modular, mobile activity monitoring system. However, all these efforts would be autotelic without giving feedback to the user, without visualizing the results of data processing and classification. Online visualization in activity monitoring applications is important to help the user to reflect on the results and gain insights about his behaviour, which then could encourage to continue or do even more physical activity. Therefore, this subsection investigates the question how to provide the user with understandable, helpful and motivating feedback. Example snapshots from the realized mobile systems are shown, visualizing results of activity recognition, intensity estimation and various heart rate-related features.

The most common visualization tools used to represent results of activity monitoring are charts (*e.g.* bars or lines) to show the time spent performing different recognized classes [18, 134]. Another way of representation is based on living metaphors, *e.g.* using a fish to look happy or sad depending on how far the user met the activity goals [97]. Moreover, Fan et al. [46] introduced the visualization tool *Spark*: They display activity data by using circles of different colour and size animated in various ways. In a field study they found that such abstract visual rewards encourage some of the test subjects to be more active as usually. A further common motivational tool is to give trophies when the user accomplishes a certain goal, *e.g.* 10.000 steps made a day (*cf.* the commercially available product Fitbit [50]).

Since the main goals of this thesis are not related to the visualization of the results, rather simple methods were implemented in the different prototypes of the mobile activity monitoring system to give feedback to the user. This feedback visualizes results of data processing and classification: what activity the user performed, for how long and with what intensity. Figure 8.1 shows the activity summary as given in the mobile application implemented on the Viliv S5 UMPC, the control unit used to record the PAMAP2 dataset. From this display the user can see a summary of his performed activities of the current day. Figure 8.1 visualizes the results of a nearly three hour session, the activity recognition and intensity estimation tasks both include the classes

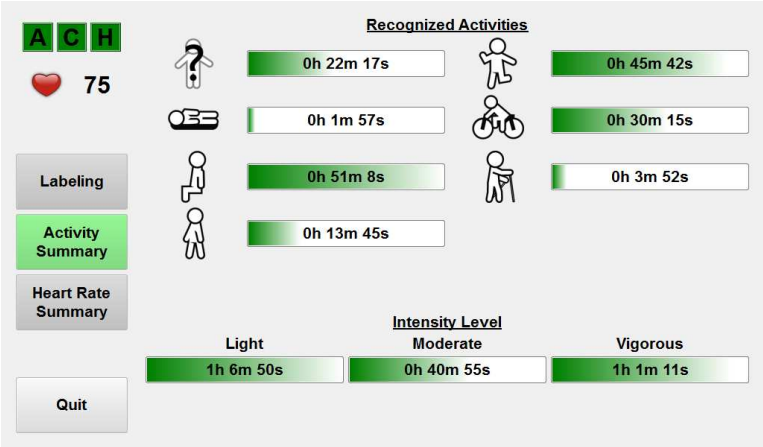


Figure 8.1: *Visualization of the data processing and classification results: Example of the activity summary on the Viliv S5 UMPC.*

as defined before (the icon with the question mark refers to the background activity class). With this online feedback the user can access his progress anywhere and at anytime, thus getting informed about *e.g.* how much more physical activity he should perform to reach the general recommendations of Haskell et al. [66]. The GUI of the final prototype of the mobile system includes a similar visualization tool. Figure 8.2 shows an example snapshot from this system, taken from the Samsung Galaxy S III smartphone.

As discussed in Section 8.2, due to the available heart rate data, additional information can be displayed for the user. On the one hand, for cardiac patients for example, a specific heart rate can be defined individually in the system, and an alarm is initiated when exceeding this value. On the other hand, a summary of how much time the user spent in different heart rate zones – these zones are based on the proposition of Fox et al. [52] and are widely used in sport applications – can be provided as well. Figure 8.3 shows this feedback given to the user in visualized form on the Viliv control unit. This heart rate summary was taken from the same session as Figure 8.1, summarizing how much time the user spent in different heart rate zones.

8.4 Integrated Activity Monitoring System

The unobtrusive monitoring of out-of-hospital physical activity, while the patient follows his regular daily routine, is an important but also difficult task. For a long time, questionnaires represented the main choice of clinical personnel, resulting in a highly imprecise control of how much physical activity the patients performed at home. However, with recent progress in wearable sensing, it becomes reasonable for individuals to wear different sensors all day, thus a more precise long-term activity monitoring is establishing.

The previous section presented a mobile and unobtrusive system that enables the accurate monitoring of physical activities in daily life. This mobile system focuses on

Figure 8.2: *Visualization of the data processing and classification results: Example of the activity summary on the Samsung Galaxy S III smartphone.*

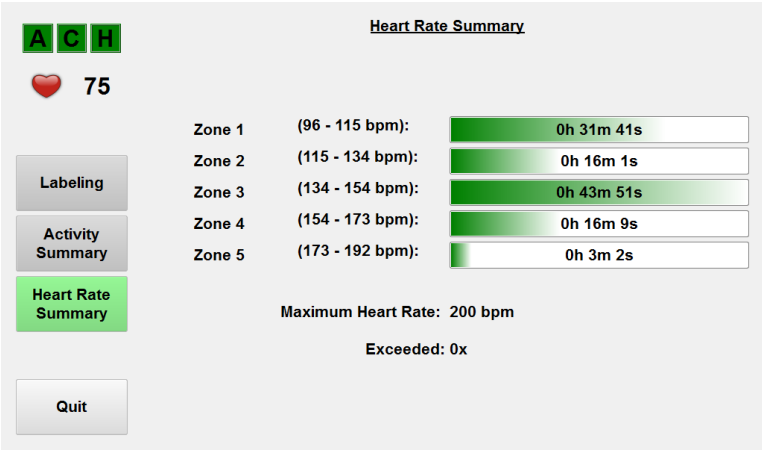


Figure 8.3: *Visualization of the features derived from the recorded heart rate information: Example of the heart rate summary on the Viliv S5 UMPC.*

the monitoring of aerobic activities, with the following two goals: 1) estimating the intensity of performed activities to be able to answer how far a patient meets the general recommendations of [66], or the goals defined in a care plan by the clinician and 2) recognizing aerobic activities traditionally recommended to give a more detailed description of a patient's daily routine. In this section the above described mobile system is integrated into a healthcare system supporting out-of-hospital services.

Various online applications for physical activity monitoring were already presented in related work (*cf. e.g.* [42]), also providing feedback to the user to preserve motivation (*e.g.* in [18]). However, the described overall system in this section is the first attempt of completely integrating such mobile systems into a professional healthcare system. The integration of the mobile platform with an Electronic Health Record (EHR) has many benefits. For example, it provides access for both the clinician (*e.g.* to enter a patient's medical record or to set up a care plan in the EHR) and the patient (*e.g.* to watch assigned educational material). It also provides valuable information to the clinical personnel to supervise program adherence and follow the patient's rehabilitation progress on daily basis. Moreover, feedback is also given to the patient about his daily progress, preserving or even increasing his motivation to follow the defined care plan.

8.4.1 System Overview

The integration of the mobile activity monitoring system with an EHR is realized during the PAMAP project [116], within the aerobic activity monitoring use case. Figure 8.4 shows the major components and their interaction in the proposed overall system for aerobic activity monitoring and support in daily life. The EHR serves for collection and management of information (related to the medical profile and history of the monitored subject, and to the collected activity information), and is further described in the next subsection. The main purpose of the mobile platform is the monitoring of the user's daily activities by collecting and processing sensory data, but it also gives an instant feedback to the user. This mobile system was described in detail in Section 8.3. The Clinician's WEB Interface provides a web based user interface for the physicians to the EHR. It enables the clinician to view and edit the medical record of the monitored subject (*cf.* Section 8.4.2), to define a personal program of aerobic activities for the subjects on daily basis, to define and upload educational material for each of his patients individually, and to view a summary of the patient's performed activities over a specific day (*cf.* Section 8.4.3). The Individual's Interactive TV (i-TV) interface provides the monitored subjects with the means to use the system's services that are offered to them. Specifically, the patient can view his own subset of the EHR, can view educational material (*e.g.* watch short videos) his clinician assigned to him, and can see the defined program of aerobic activities for the current day. The i-TV provides hereby a convenient interface even for subjects – especially for elderly – who are not very familiar with computers, since it can be controlled with a standard TV remote control.

A typical scenario of using the system by the clinician and a new patient – interacting with the different components of the system – is described in the following

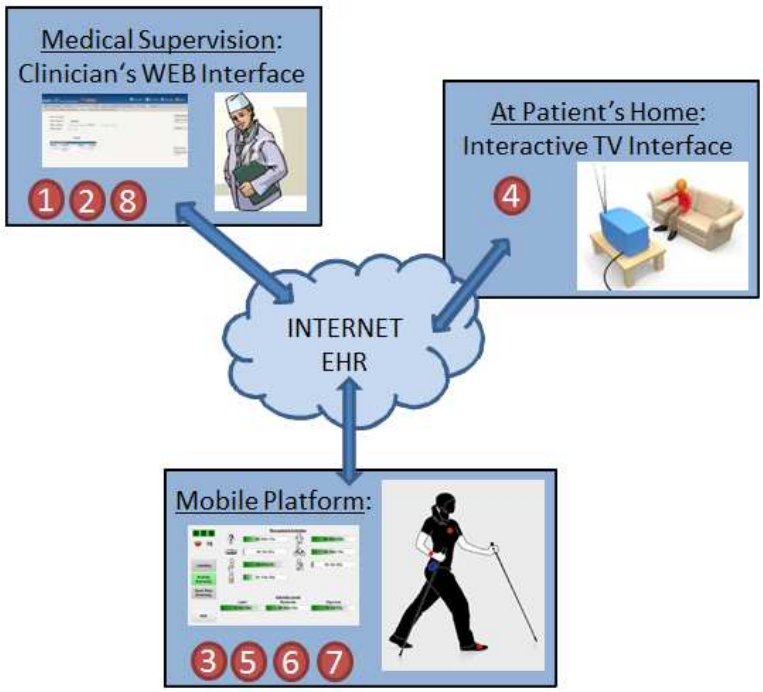


Figure 8.4: *The integration of the mobile physical activity monitoring system into a complete healthcare system. The figure shows the major components and their interaction for aerobic activity monitoring and support in daily life.*

steps (steps 1-3 are only carried out at the setup for a new patient, while steps 4-8 are performed every day, the numbers in the major components of Figure 8.4 refer to these steps):

1. The clinician adds (registers) the new patient in the EHR, and enters information about the new patient, related to his medical profile and history.

2. The clinician draws up the care plan to be followed by the patient, and enters it into the EHR. This care plan includes a set of measurements to be performed periodically, a set of questionnaires to be filled out, a set of educational material to inform the patient, etc. The care plan also defines a personal program of aerobic activities to be followed by the new patient.

3. The clinician downloads basic personal information (age, resting heart rate, etc.) of the new patient into the mobile platform (before first used by the patient), using the mobile application (done automatically after corresponding button pressed). This personal data is used for the computation of personalized features (*cf.* Section 4.2.3), and defines parameters for the heart rate summary screen (HR-zones and maximum HR, *cf.* Section 8.3.3 and Figure 8.3).

4. At home, the patient informs himself in the morning about the current day's assigned activity program, using the i-TV interface.

5. The patient wears the mobile platform (*cf.* Section 8.3) over the active part of the day. The mobile application records and processes the sensory data.

6. The patient can look at any time at the mobile application to see his progress. The mobile application's GUI gives feedback for the monitored subject, as described in Section 8.3.3.

7. At the end of the day, the patient uploads the result of the activity monitoring to the EHR, using the mobile application (done automatically after corresponding button pressed).

8. The clinician can look at the patient's daily progress using the visualization provided in the web interface of the EHR (*cf.* Section 8.4.3), thus supervising how far the patient followed the defined program. If not sufficiently, or the program has to be readjusted, the clinician can contact his patient.

8.4.2 Electronic Health Record

The Electronic Health Record stores a comprehensive summary of the medical record of the monitored patient, and stores collected activity information. The stored medical record includes a general health profile of the patient (family health record, habits and social history – *e.g.* smoking, alcohol consumption – allergies, vaccinations, body mass index, etc.), a history of the patient's visits, results of laboratory and other medical tests, diagnoses, medications, surgeries, and the care plan definition. The Clinician's WEB Interface enables healthcare professionals to view and edit the patient's medical record in the EHR. Figure 8.5 for example presents the screen where the clinician can view and edit the health related habits of living. Furthermore, the patient can view his own subset of medical record at home using the i-TV interface. This also enables the patient to watch assigned educational material, and to fill out different questionnaires in defined intervals (daily, weekly, etc).

The collected activity information of monitored subjects (including results of estimated intensity and recognized activity) is also stored in the EHR. A binary message format was defined for effective data communication and storage. Each message includes a timestamp and information about estimated intensity, recognized activity and measured heart rate (thus aside from heart rate, not raw sensory data but only the result after data processing is stored). The size of one message is only 27 bytes. The activity and intensity information is smoothed in the mobile application, generating only one message every 30 seconds. These messages are collected during one day and bundled into one binary file by the mobile application, and then sent to the EHR at the end of the day. Since the subject wears the mobile platform up to six hours per day (this limit is set by the battery time of the hardware components of the mobile platform, but is usually sufficient to cover the active part of the day), the size of the binary file (containing all generated binary messages from the particular day) is only approximately 19kB. The communication between the EHR and the mobile application is based on the web services provided by the EHR's API. Given the EHR's transaction-URL and the respective patient's identifier in the EHR, the binary file can be sent at the end of the day for storage from the mobile application to the

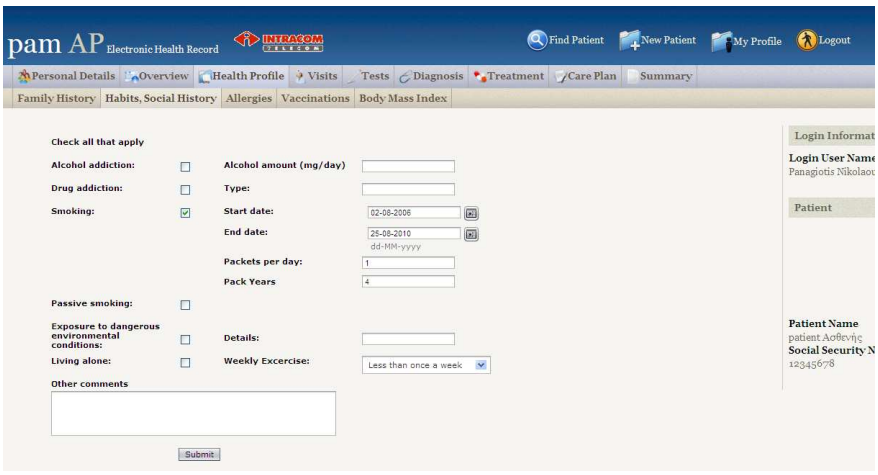


Figure 8.5: *Example screen of the Electronic Health Record: health related habits of living.*

EHR (step 7 in the above described scenario), or personal information can be queried by the mobile application from the EHR (step 3 in the scenario). Both of these actions are predefined tasks in the mobile application and can therefore be easily executed by pressing the corresponding button.

8.4.3 Evaluation of the Integrated Overall System

The clinical trial already mentioned in Section 8.3 was also used to test and evaluate the entire integrated system (all the steps 1-8 as described in Section 8.4.1). The goal was to confirm that each of the components works well, and their interaction is error-free. For all of the 30 elderly subjects participating in this study the 8-step scenario was followed, starting with steps 1-3 at the beginning and performing steps 4-8 every day over one week per participant.

The result of the clinical trial, from the participant's point of view was that they could use the different components of the overall system accessible for them without major issues. It should be noticed that – since the clinical trial was carried out in France [116] – the user interface of the mobile application is multilingual. At the time of the trials it supported three languages: English, French and German. From the clinician' point of view, accessing EHR, setting up the system for the patients, etc. was also without major issues. Moreover, not only the participants could see their progress given on the mobile device, but the clinician could also access this progress on a daily basis. Figure 8.6 shows an example: The meaningful part of an elderly subject's daily recording session (including housework activities, a long walking, cooking and eating), as shown for the clinician in the web interface of the EHR.

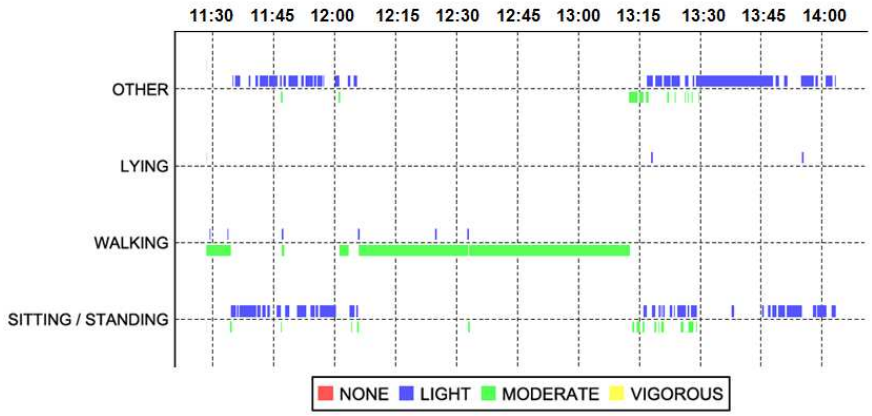


Figure 8.6: *The clinician's feedback about the patient's daily progress, as provided in the integrated system: Example activity summary as shown in the EHR's web interface.*

8.5 Conclusion

This chapter presented a modular, mobile activity monitoring system integrated in a clinical application, supporting the recognition and estimation of aerobic activities and out-of-hospital services. The major components of the overall system are an EHR, accessible by the patient via the i-TV interface and by the clinician via a web interface, and a mobile platform. The proposed mobile system consists of small, low power wireless sensors (3 accelerometers and a heart rate monitor) and an Android smartphone. The different methods and algorithms introduced and described in the previous chapters for recognizing physical activities and estimating their intensity level were implemented here, thus realizing an accurate, robust and unobtrusive physical activity monitoring system for daily life.

Experiments in Section 8.2 with various sets of sensors justified the idea of a modular activity monitoring system, since different sets are required for different tasks. Moreover, empirical studies in Section 8.3 showed that more complex, meta-level classifiers (boosted decision trees as a concrete example) are feasible and thus a considerable choice for mobile applications, there are no limitations regarding the computational costs. However, it should be noted that more complex classifiers lead to more energy consumption, thus could shorten the batter time of the mobile system's hardware components.

A further important topic of this chapter was the feedback given based on the data processing and classification results: The proposed integrated system makes it possible to visualize and review the daily activities of the patients. For the patient, this preserves or even increases the motivation to follow a defined care plan. For the clinical personnel, it provides valuable information on program adherence, and lightens the estimation of rehabilitation progress. Examples shown in *e.g.* Figure 8.1 or Figure 8.6 justified the implementation of the previously proposed methods. However, these results also exposed some challenges and thus the further need for im-

provement. For example, the clinical trials with 30 elderly subjects revealed some weaknesses when dealing with different houseworking activities. Therefore, a more advanced post-processing step or the introduction of high-level activity recognition should be investigated in future work.

9

Conclusion

The main goal defined for this thesis was the development of a mobile, personalized physical activity monitoring system applicable for everyday life scenarios. The goal was motivated by the fact that regular physical activity is essential to maintain or even improve an individual's health. It is important to monitor how much physical activity individuals do during their daily routine, to be able to tell how far they meet professional recommendations. Such recommendations or general guidelines exist for all the different age groups to perform aerobic, muscle-strengthening, flexibility or balance exercises. From these recommendations, this thesis concentrated on monitoring aerobic physical activity. Two main objectives were defined in this context. On the one hand, the goal was to estimate the intensity of performed activities: To distinguish activities of light, moderate or vigorous effort. On the other hand, the goal was also to recognize basic aerobic activities (such as *walk*, *run* or *cycle*) and basic postures (*lie*, *sit* and *stand*). This way, the developed system can give a more detailed description of an individual's daily routine.

9.1 Results

The hardware already exist to create the desired physical activity monitoring system in an unobtrusive way, *e.g.* by using current smart phone technology and wearable sensors. Therefore, the focus of this thesis was on the development of methods for physical activity recognition and intensity estimation, which are applicable for the envisioned mobile system. Emphasis was placed thereby on identifying key challenges in this research field and on addressing them with the introduction of novel methods and algorithms. Moreover, it should be noted that a high value was put on the evaluation of the proposed methods: Thorough experiments are presented in the respective chapters of this thesis to justify the introduced data processing and classification methods.

The major contributions presented in this thesis are the following:

- Creation of two new datasets for physical activity monitoring, including a wide range of physical activities. Moreover, both datasets have been made publicly available and can already show a certain impact in the research community.
- Benchmark of various activity recognition and intensity estimation problems with commonly used classification algorithms.
- Investigation of the means to create robust activity monitoring systems for everyday life, which includes the concept and modeling of other activities and highlighting the importance of subject independent validation techniques.
- Introduction of a new evaluation technique (called leave-one-activity-out) to simulate when performing previously unknown activities in a physical activity monitoring system.
- Introduction and validation of a confidence-based extension of the well known AdaBoost.M1 algorithm, called ConfAdaBoost.M1.
- Introduction and validation of a novel general concept for the personalization of physical activity recognition applications.
- Introduction and validation of a novel algorithm (called Dependent Experts), based on the concept of weighted majority voting.
- Presentation of the idea of a modular activity monitoring system, where different sets of sensors are required for different activity recognition and intensity estimation tasks.
- Integration of the developed mobile system into a full healthcare application for aerobic activity monitoring and support in daily life.

The listed contributions are both of theoretical (*cf. e.g.* the novel algorithms and developed models) and of practical value (*cf. e.g.* the proposed evaluation techniques). Some of the contributions are directly benefiting the research community (*e.g.* the created and benchmarked datasets). Moreover, this thesis also deals with the implementation of the presented methods, in order to realize the envisioned mobile system for physical activity monitoring.

9.2 Future Work

The contributions of this thesis can be understood as different important steps towards creating a mobile, unobtrusive physical activity monitoring system for everyday life. However, some questions still remain to be answered in order to completely realize the overall goal. In the following paragraphs a few important next steps are outlined, which give possible future directions to continue and extend the work presented in this thesis.

Extensive Data Collection. Although this thesis introduced two large datasets, these datasets still show some shortcomings. Two of the important limitations are the similar set of users (considering *e.g.* age or physical fitness) and the only semi-naturalistic data collection protocol. However, with the mobile system presented in Section 8.3.1 a robust and unobtrusive tool is provided to perform further data recordings. A new dataset of physical activities should include subjects from all the different age groups: children, adolescents, young and middle-aged adults and elderly. Moreover, further user groups should be included as well, *e.g.* people with overweight or with disabilities. Such a dataset would provide an excellent basis to further evaluate and improve the personalization approaches proposed in Chapter 7. Another important aspect of a new dataset should be to at least partially record it under realistic conditions, thus during the subjects' regular daily routine. This would also enable to test *e.g.* the user acceptance of the mobile system, and would provide data for high-level activity recognition.

High-Level Activity Recognition. This thesis presented numerous methods related to low-level activity recognition (the recognition of activities such as *sit*, *stand*, *walk* or *drive a car*), overall very accurate results were achieved on this topic. However, some difficulties were exposed *e.g.* during the PAMAP clinical trials (*cf.* Chapter 8) while different houseworking activities were performed, or generally when dealing with the composite activity *play soccer*. A promising way to overcome these difficulties is to investigate methods for high-level or composite activity recognition (the recognition of activities such as *going shopping*, *preparing food* or *eating dinner*), as done *e.g.* in [19, 78, 79]. For this purpose temporal information should also be taken into account, since patterns in the order of performed activities exist in real-life situations. Therefore, methods such as hidden Markov models should be considered to model this behaviour.

Semi-supervised Learning. The methods presented in this thesis all rely on only annotated data. However, as discussed in Chapter 3, obtaining ground truth for recorded sensory data is not straightforward. With the available technology it is easy to generate large datasets nowadays, but labeling still requires expensive human effort. Therefore, semi-supervised learning receives increasing attention in the machine learning community [204]. These methods can combine a small amount of labeled data with large amounts of unlabeled data. Semi-supervised learning methods have been applied in the physical activity monitoring research field recently, delivering promising results [5, 37, 76]. A special case of semi-supervised learning is active learning, where the learning algorithm chooses the most informative data samples to be annotated [159]. An application of this approach for human activity recognition was shown by Alemdar et al. [3]. Therefore, and considering the above described plan for a new extensive data collection, semi-supervised learning methods deserve further attention.

Extension of the Modular Activity Monitoring System. Two sensors have been investigated in this thesis: accelerometer and heart rate monitor. Building on these two sensors, a modular mobile activity monitoring system was presented in Chapter 8.

By using different subsets of the available sensors, different functionality is provided by the modular system. However, the presented system could be extended with additional sensors, in order to increase the accuracy of the already provided functions and to extend the system with new functionality. One possible extension is to add further physiological sensors, such as GSR (galvanic skin response) or ECG (electrocardiogram), both available in the Shimmer platform [160]. On the one hand, with these additional sensors the accuracy of especially the assessment of the performed activities' intensity level can be further improved. On the other hand, new functionality related to physiological variables (stress, tiredness, etc.) could be added to the modular system [119]. Examples of using physiological sensors for mental stress detection [170, 192] or assessing cognitive load [64] have been presented recently. Another possible way to extend the modular system is to add location information, *e.g.* a GPS sensor. This would on the one hand increase the accuracy of activity recognition, especially related to high-level activities. On the other hand, numerous new functions would be available on the mobile system related to *e.g.* navigation. This in turn would further increase the applicability of the system, and thus would further motivate individuals to use such applications in their everyday life.

A

Abbreviations and Acronyms

Abbreviation	Meaning
ADL	Activities of daily living
ANN	Artificial neural network
ARC	Activity recognition chain
BMI	Body mass index
BPM	Beats per minute
CRF	Conditional random field
CV	Cross-validation
DE	Dependent experts
DFAR	Device-free radio-based activity recognition
DFT	Discrete Fourier transform
DOF	Degrees of freedom
DPC	Data processing chain
DT	Decision tree
ECG	Electrocardiogram
ECOC	Error-correcting output codes
EHR	Electronic health record
EMG	Electromyogram
FFT	Fast Fourier transform
GA	Genetic algorithm
GAMBLE	Gentle adaptive multiclass boosting learning
GPS	Global positioning system
GSR	Galvanic skin response
GUI	Graphical user interface
HMM	Hidden Markov model
HR	Heart rate
IADL	Instrumental activities of daily living

Abbreviation	Meaning
ICDM	International Conference on Data Mining
IMU	Inertial measurement unit
ISWC	International Symposium on Wearable Computers
KNN	k-Nearest Neighbor
LOAO	Leave-one-activity-out
LOOAO	Leave-one-other-activity-out
LOSO	Leave-one-subject-out
MEMS	Micro-electro-mechanical system
MET	Metabolic equivalent
MFCC	Mel-frequency cepstral coefficient
MHR	Maximum heart rate
MV	Majority Voting
PAMAP	Physical activity monitoring for aging people
PCA	Principal component analysis
PSD	Power spectral density
RFID	Radio-frequency identification
RSSI	Received signal strength indicator
RWMA	Randomized weighted majority algorithm
SAMME	Stagewise additive modeling using a multi-class exponential loss function
SVM	Support Vector Machine
UMPC	Ultra-mobile personal computer
WEKA	Waikato Environment for Knowledge Analysis
WHO	World Health Organization
WMA	Weighted majority algorithm
WMV	Weighted majority voting

B

Datasets: Supplementary Material

This appendix presents supplementary material related to the PAMAP and PAMAP2 datasets, both described in Chapter 3.

Table B.1: *Data format of the published PAMAP dataset. The data files contain 45 columns, described on the left side. The right side of the table specifies the content of an IMU sensor (hand, chest or foot) data.*

Column	Data content	Column	Data content
1	timestamp (s)	1	temperature (°C)
2	activity ID	2-4	3D-accelerometer (ms^{-2})
3	heart rate (bpm)	5-7	3D-gyroscope (°/s)
4-17	IMU hand	8-10	3D-magnetometer (μT)
18-31	IMU chest	11-14	orientation (turned off)
32-45	IMU foot		

Table B.2: *Data format of the published PAMAP2 dataset. The data files contain 54 columns, described on the left side. The right side of the table specifies the content of an IMU sensor (hand, chest or ankle) data.*

Column	Data content	Column	Data content
1	timestamp (s)	1	temperature (°C)
2	activity ID	2-4	3D-accelerometer (ms^{-2}), scale: ±16g
3	heart rate (bpm)	5-7	3D-accelerometer (ms^{-2}), scale: ±6g
4-20	IMU hand	8-10	3D-gyroscope (°/s)
21-37	IMU chest	11-13	3D-magnetometer (μT)
38-54	IMU ankle	14-17	orientation (turned off)

Table B.3: *PAMAP dataset: detailed information on the participating subjects.*

Subject ID	Sex	Age (years)	Height (cm)	Weight (kg)	Resting HR (bpm)	Dominant hand
subject1	female	29	175	51	47	right
subject2	male	27	182	92	67	right
subject3	male	30	168	62	56	right
subject4	male	31	193	85	54	right
subject5	male	25	180	70	69	right
subject6	male	26	181	75	59	left
subject7	male	29	174	91	56	right
subject8	male	26	182	85	63	right

Table B.4: *PAMAP2 dataset: detailed information on the participating subjects.*

Subject ID	Sex	Age (years)	Height (cm)	Weight (kg)	Resting HR (bpm)	Dominant hand
101	male	27	182	83	75	right
102	female	25	169	78	74	right
103	male	31	187	92	68	right
104	male	24	194	95	58	right
105	male	26	180	73	70	right
106	male	26	183	69	60	right
107	male	23	173	86	60	right
108	male	32	179	87	66	left
109	male	31	168	65	54	right

Table B.5: *Brief description of the* 14 *different performed activities, included in the PAMAP dataset.*

Activity	Description
lying	lying quietly while doing nothing, small movements (*e.g.* changing the lying posture) are allowed
sitting	sitting in a chair, mainly consisting of working with a computer
standing	consists of standing still or standing still and talking, possibly gesticulating
ironing	consists of ironing and folding one or two shirts
vacuum cleaning	vacuum cleaning one or two office rooms, including moving objects (*e.g.* chairs) placed on the floor
ascending stairs	performed in a building between the ground and the top floors, a distance of five floors had to be covered going upstairs
descending stairs	performed in a building between the top and the ground floors, a distance of five floors had to be covered going downstairs
very slow walking	walking outside with a speed of less than 3 kmh^{-1}
normal walking	walking outside with moderate to brisk pace with a speed of $4 - 6$ kmh^{-1}, according to what was suitable for the subject
Nordic walking	performed outside on asphaltic terrain, using asphalt pads on the walking poles (it has to be noted that none of the subjects was very familiar with this sport activity)
running	jogging outside with a suitable speed for each subject
cycling	performed outside with slow to moderate pace, as if the subject would bike to work or bike for pleasure (but not as a sport activity)
playing soccer	subject played soccer with the supervisor, which mainly consisted of running with the ball, dribbling, passing the ball to the supervisor or shooting the ball
rope jumping	the subjects used the technique most suitable for them, which mainly consisted of the basic jump (where both feet jump at the same time over the rope) or the alternate foot jump (where alternate feet are used to jump off the ground)

Table B.6: *Brief description of the* 18 *different performed activities, included in the PAMAP2 datasets.*

Activity	Description
lying	lying quietly while doing nothing, small movements (*e.g.* changing the lying posture) are allowed
sitting	sitting in a chair in whatever posture the subject feels comfortable, changing the sitting postures is allowed
standing	consists of standing still or standing still and talking, possibly gesticulating
ironing	ironing 1 – 2 shirts or T-shirts
vacuum cleaning	vacuum cleaning one or two office rooms, including moving objects (*e.g.* chairs) placed on the floor
ascending stairs	performed in a building between the ground and the top floors, a distance of five floors had to be covered going upstairs
descending stairs	performed in a building between the top and the ground floors, a distance of five floors had to be covered going downstairs
normal walking	walking outside with moderate to brisk pace with a speed of $4 - 6$ kmh^{-1}, according to what was suitable for the subject
Nordic walking	performed outside on asphaltic terrain, using asphalt pads on the walking poles (it has to be noted that none of the subjects was very familiar with this sport activity)
cycling	performed outside with slow to moderate pace, as if the subject would bike to work or bike for pleasure (but not as a sport activity)
running	jogging outside with a suitable speed for each subject
rope jumping	the subjects used the technique most suitable for them, which mainly consisted of the basic jump (where both feet jump at the same time over the rope) or the alternate foot jump (where alternate feet are used to jump off the ground)
watching TV	watching TV at home in whatever posture (lying, sitting) the subject feels comfortable
computer work	working in front of a PC at the office
car driving	driving in the city between the office and the subject's home
folding laundry	folding shirts, T-shirts and/or bed linens
house cleaning	dusting some shelves, including removing books and other things and putting them back onto the shelves
playing soccer	playing 1 vs. 1 or 2 vs. 1, including running with the ball, dribbling, passing the ball and shooting the ball on goal

Bibliography

[1] Barbara E. Ainsworth, William L. Haskell, Melicia C. Whitt, Melinda L. Irwin, Ann M. Swartz, Scott J. Strath, William L. O'Brien, David R. Bassett, Kathryn H. Schmitz, Patricia O. Emplaincourt, David R. Jacobs, and Arthur S. Leon. Compendium of physical activities: an update of activity codes and MET intensities. *Medicine and Science in Sports and Exercise*, 32(9):498–516, September 2000.

[2] Fahd Albinali, Stephen S. Intille, William L. Haskell, and Mary Rosenberger. Using wearable activity type detection to improve physical activity energy expenditure estimation. In *Proceedings of 12th International Conference on Ubiquitous Computing (UbiComp)*, pages 311–320, Copenhagen, Denmark, September 2010.

[3] Hande Alemdar, Tim L. M. van Kasteren, and Cem Ersoy. Using active learning to allow activity recognition on a large scale. In *Proceedings of 2nd International Conference on Ambient Intelligence (AmI)*, pages 105–114, Amsterdam, Netherlands, November 2011.

[4] Leslie Alford. What men should know about the impact of physical activity on their health. *International Journal of Clinical Practice*, 64(13):1731–1734, December 2010.

[5] Aziah Ali, Rachel C. King, and Guang-Zhong Yang. Semi-supervised segmentation for activity recognition with Multiple Eigenspaces. In *Proceedings of 5th International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pages 314–317, Hong Kong, China, June 2008.

[6] Fevzi Alimoglu and Ethem Alpaydin. Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition. In *Proceedings of 5th Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN)*, Istanbul, Turkey, 1996.

[7] Bashar Altakouri, Gerd Kortuem, Agnes Grünerbl, Kai Kunze, and Paul Lukowicz. The benefit of activity recognition for mobile phone based nursing documentation: a Wizard-of-Oz study. In *Proceedings of IEEE 14th International Symposium on Wearable Computers (ISWC)*, Seoul, South Korea, October 2010.

[8] Kerem Altun, Billur Barshan, and Orkun Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, October 2010.

[9] Ran Avnimelech and Nathan Intrator. Boosting regression estimators. *Neural Computation*, 11(2):499–520, February 1999.

[10] Oresti Baños, Miguel Damas, Héctor Pomares, Ignacio Rojas, Máté A. Tóth, and Oliver Amft. A benchmark dataset to evaluate sensor displacement in activity recognition. In *Proceedings of 14th International Conference on Ubiquitous Computing (UbiComp)*, pages 1026–1035, Pittsburgh, PA, USA, September 2012.

[11] Oresti Baños, Miguel Damas, Héctor Pomares, Fernando Rojas, Blanca Delgado-Marquez, and Olga Valenzuela. Human activity recognition based on a sensor weighting hierarchical classifier. *Soft Computing*, 17(2):333–343, February 2013.

[12] Kevin Bache and Moshe Lichman. UCI Machine Learning Repository. URL `http://archive.ics.uci.edu/ml`.

[13] Marc Bächlin, Martin Kusserow, Hanspeter Gubelmann, and Gerhard Tröster. Ski jump analysis of an Olympic champion with wearable acceleration sensors. In *Proceedings of IEEE 14th International Symposium on Wearable Computers (ISWC)*, Seoul, South Korea, October 2010.

[14] Gernot Bahle, Kai Kunze, Koichi Kise, and Paul Lukowicz. I see you: How to improve wearable activity recognition by leveraging information from environmental cameras. In *Proceedings of 11th IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 409–412, San Diego, CA, USA, March 2013.

[15] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In *Proceedings of 2nd International Conference on Pervasive Computing (PERVASIVE)*, pages 1–17, Linz/Vienna, Austria, April 2004.

[16] Martin Berchtold, Matthias Budde, Dawud Gordon, Hedda R. Schmidtke, and Michael Beigl. ActiServ: Activity recognition service for mobile phones. In *Proceedings of IEEE 14th International Symposium on Wearable Computers (ISWC)*, Seoul, South Korea, October 2010.

[17] Gerald Bieber and Christian Peter. Using physical activity for user behavior analysis. In *Proceedings of 1st International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, Athens, Greece, July 2008.

[18] Gerald Bieber, Jörg Voskamp, and Bodo Urban. Activity recognition for everyday life on mobile phones. In *Proceedings of 5th International Conference on Universal Access in Human-Computer Interaction (UAHCI)*, pages 289–296, San Diego, CA, USA, July 2009.

[19] Ulf Blanke and Bernt Schiele. Remember and transfer what you have learned - recognizing composite activities based on activity spotting. In *Proceedings of IEEE 14th International Symposium on Wearable Computers (ISWC)*, Seoul, South Korea, October 2010.

[20] Avrim Blum. On-line algorithms in machine learning. In *Proceedings of Workshop on On-Line Algorithms, Dagstuhl*, pages 306–325, Dagstuhl, Germany, June 1996.

[21] BM-innovations. BM innovations GmbH development and product website, 2013-09-16. URL `http://www.bm-innovations.com`.

[22] Alberto G. Bonomi, Guy Plasqui, Annelies Goris, and Klaas R. Westerterp. Improving assessment of daily energy expenditure by identifying types of physical activity with a single accelerometer. *Journal of Applied Physiology*, 107(3): 655–661, September 2009.

[23] Marko Borazio and Kristof van Laerhoven. Combining wearable and environmental sensing into an unobtrusive tool for long-term sleep studies. In *Proceedings of 2nd ACM SIGHIT International Health Informatics Symposium (IHI)*, pages 71–80, Miami, FL, USA, January 2012.

[24] Alan K. Bourke and Gerald M. Lyons. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering & Physics*, 30(1): 84–90, January 2008.

[25] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996.

[26] Leah Buechley. A construction kit for electronic textiles. In *Proceedings of IEEE 10th International Symposium on Wearable Computers (ISWC)*, pages 83–90, Montreux, Switzerland, October 2006.

[27] Andrew Campbell and Tanzeem Choudhury. From smart to cognitive phones. *IEEE Pervasive Computing*, 11(3):7–11, March 2012.

[28] Hong Cao, Minh Nhut Nguyen, Clifton Phua, Shonali Krishnaswamy, and Xiao-Li Li. An integrated framework for human activity classication. In *Proceedings of 14th International Conference on Ubiquitous Computing (UbiComp)*, pages 331–340, Pittsburgh, PA, USA, September 2012.

[29] Kuang-I Chang, Yen-Hsien Lee, Yu-Jen Su, Hong-Dun Lin, and Bor-Nian Chuang. Portable driver drowsiness prediction device and method. In *Proceedings of 33rd Annual International IEEE EMBS Conference*, pages 4390–4393, Boston, MA, USA, August-September 2011.

[30] Chao Chen, Daqing Zhang, Lin Sun, Mossaab Hariz, and Yang Yuan. Does location help daily activity recognition? In *Proceedings of 10th International Conference on Smart Homes and Health Telematics (ICOST)*, pages 83–90, Artimino, Italy, June 2012.

[31] Heng-Tze Cheng, Martin Griss, Paul Davis, Jianguo Li, and Di You. Toward zero-shot learning for human activity recognition using semantic attribute sequence model. In *Proceedings of 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pages 355–358, Zurich, Switzerland, September 2013.

[32] David A. Clifton, Lei Clifton, Samuel Hugueny, David Wong, and Lionel Tarassenko. An extreme function theory for novelty detection. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):28–37, February 2013.

[33] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y. Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, Ian Smith, and James A. Landay. Activity sensing in the wild: a field trial of ubifit garden. In *Proceedings of 26th SIGCHI Conference on Human Factors in Computing Systems*, pages 1797–1806, Florence, Italy, April 2008.

[34] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

[35] Scott E. Crouter, Kurt G. Clowers, and David R. Bassett. A novel method for using accelerometer data to predict energy expenditure. *Journal of Applied Physiology*, 100(4):1324–1331, April 2006.

[36] Scott E. Crouter, James R. Churilla, and David R. Bassett. Accuracy of the Actiheart for the assessment of energy expenditure in adults. *European Journal of Clinical Nutrition*, 62(6):704–711, June 2008.

[37] Božidara Cvetković, Mitja Luštrek, Boštjan Kaluža, and Matjaž Gams. Semi-supervised learning for adaptation of human activity recognition classifier to the user. In *Proceedings of 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Spain, July 2011.

[38] Anthony Dalton and Gerald OLaighin. Comparing supervised learning techniques on the task of physical activity recognition. *IEEE Transactions on Information Technology in Biomedicine*, 2012.

[39] Jakob Doppler, Gerald Holl, Alois Ferscha, Marquart Franz, Cornel Klein, Marcos Dos Santos Rocha, and Andreas Zeidler. Variability in foot-worn sensor placement for activity recognition. In *Proceedings of IEEE 13th International Symposium on Wearable Computers (ISWC)*, pages 143–144, Linz, Austria, September 2009.

[40] Thi V. Duong, Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-Markov model. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 838–845, San Diego, CA, USA, June 2005.

[41] Günther Eibl and Karl P. Pfeiffer. How to make AdaBoost.M1 work for weak base classifiers by changing only one line of the code. In *Proceedings of 13th European Conference on Machine Learning (ECML)*, pages 72–83, Helsinki, Finland, August 2002.

[42] Miikka Ermes, Juha Pärkkä, and Luc Cluitmans. Advancing from offline to online activity recognition with wearable sensors. In *Proceedings of 30th Annual International IEEE EMBS Conference*, pages 4451–4454, Vancouver, Canada, August 2008.

[43] Miikka Ermes, Juha Pärkkä, Jani Mäntyjärvi, and Ilkka Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Transactions on Information Technology in Biomedicine*, 12 (1):20–26, January 2008.

[44] EZ430-Chronos. eZ430-Chronos Texas Instruments Wiki, 2013-09-16. URL `http://processors.wiki.ti.com/index.php/EZ430-Chronos`.

[45] Jean Baptiste Faddoul, Boris Chidlovskii, Rémi Gilleron, and Fabien Torre. Learning multiple tasks with boosted decision trees. In *Proceedings of 2012 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, pages 681–696, Bristol, UK, September 2012.

[46] Chloe Fan, Jodi Forlizzi, and Anind K. Dey. A spark of activity: exploring informative art as visualization for physical activity. In *Proceedings of 14th International Conference on Ubiquitous Computing (UbiComp)*, pages 81–84, Pittsburgh, PA, USA, September 2012.

[47] Jesus Favela, Monica Tentori, Luis A. Castro, Victor M. Gonzalez, Elisa B. Moran, and Ana I. Martínez-García. Activity recognition for context-aware hospital applications: issues and opportunities for the deployment of pervasive networks. *Mobile Networks and Applications*, 12(2-3):155–171, March 2007.

[48] Davide Figo, Pedro C. Diniz, Diogo R. Ferreira, and João M.P. Cardoso. Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing*, 14(7):645–662, October 2010.

[49] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[50] FITBIT. Fitbit product website, 2013-09-16. URL `http://www.fitbit.com`.

[51] Evelyn Fix and J. L. Hodges. Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report 4, US Air Force School of Aviation Medicine, Randolph Field, Texas, February 1951.

[52] Samuel M. Fox, John P. Naughton, and William L. Haskell. Physical activity and the prevention of coronary heart disease. *Annals of Clinical Research*, 3 (6):404–432, December 1971.

[53] Korbinian Frank, Maria Josefa Vera Nadales, Patrick Robertson, and Tom Pfeifer. Bayesian recognition of motion related activities with inertial sensors. In *Proceedings of 12th International Conference on Ubiquitous Computing (UbiComp)*, pages 445–446, Copenhagen, Denmark, September 2010.

[54] Yoav Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, September 1995.

[55] Yoav Freund and Robert E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[56] Yoav Freund and Robert E. Schapire. Response to Mease and Wyner, Evidence contrary to the statistical view of boosting, JMLR 9:131-156, 2008. *Journal of Machine Learning Research*, 9:171–174, June 2008.

[57] Peter W. Frey and David J. Slate. Letter recognition using Holland-style adaptive classifiers. *Machine Learning*, 6(2):161–182, March 1991.

[58] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.

[59] Yuichi Fujiki, Konstantinos Kazakos, Colin Puri, Pradeep Buddharaju, Ioannis Pavlidis, and James Levine. NEAT-o-Games: blending physical activity and fun in the daily routine. *Computers in Entertainment*, 6(2), July 2008.

[60] Vanessa Gómez-Verdejo, Manuel Ortega-Moral, Jerónimo Arenas-García, and Aníbal R. Figueiras-Vidal. Boosting by weighting critical and erroneous samples. *Neurocomputing*, 69(7-9):679–685, March 2006.

[61] Dawud Gordon, Hedda Rahel Schmidtke, Michael Beigl, and Georg Von Zengen. A novel micro-vibration sensor for activity recognition: potential and limitations. In *Proceedings of IEEE 14th International Symposium on Wearable Computers (ISWC)*, Seoul, South Korea, October 2010.

[62] Dawud Gordon, Jürgen Czerny, Takashi Miyaki, and Michael Beigl. Energy-efficient activity recognition using prediction. In *Proceedings of IEEE 16th International Symposium on Wearable Computers (ISWC)*, pages 29–36, Newcastle, UK, June 2012.

[63] Norbert Győrbíró, Ákos Fábián, and Gergely Hományi. An activity recognition system for mobile phones. *Mobile Networks and Applications*, 14(1):82–91, February 2009.

[64] Eija Haapalainen, SeungJun Kim, Jodi F. Forlizzi, and Anind K. Dey. Psychophysiological measures for assessing cognitive load. In *Proceedings of 12th International Conference on Ubiquitous Computing (UbiComp)*, pages 301–310, Copenhagen, Denmark, September 2010.

[65] Mark A. Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11(1):10–18, November 2009.

[66] William L. Haskell, I-Min Lee, Russell R. Pate, Kenneth E. Powell, Steven N. Blair, Barry A. Franklin, Caroline A. Macera, Gregory W. Heath, Paul D. Thompson, and Adrian Bauman. Physical activity and public health: Updated recommendation for adults from the American College of Sports Medicine and the American Heart Association. *Medicine and Science in Sports and Exercise*, 39 (8):1423–34, August 2007.

[67] Yi He, Ye Li, and Shu-Di Bao. Fall detection by built-in tri-accelerometer of smartphone. In *Proceedings of IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 184–187, Hong Kong, China, January 2012.

[68] Tomoya Hirano and Takuya Maekawa. A hybrid unsupervised/supervised model for group activity recognition. In *Proceedings of 17th Annual International Symposium on Wearable Computers (ISWC)*, pages 21–24, Zurich, Switzerland, September 2013.

[69] Gerold Hölzl, Marc Kurz, and Alois Ferscha. Goal oriented opportunistic recognition of high-level composed activities using dynamically configured hidden Markov models. In *Proceedings of 3rd International Conference on Ambient Systems, Networks and Technologies (ANT)*, pages 308–315, Niagara Falls, ON, Canada, August 2012.

[70] Chih-wei Hsu, Chih-chung Chang, and Chich-jen Lin. A practical guide to support vector classification. *Bioinformatics*, 1(1):1–16, 2010.

[71] Bing Hu, Yanping Chen, and Eamonn J. Keogh. Time series classification under more realistic assumptions. In *SIAM Conference on Data Mining (SDM)*, Austin, TX, USA, May 2013.

[72] Jian Huang, Seyda Ertekin, Yang Song, Hongyuan Zha, and C. Lee Giles. Efficient multiclass boosting classification with active learning. In *SIAM International Conference on Data Mining (SDM)*, Minneapolis, MN, USA, April 2007.

[73] Tzu-Kuo Huang and Jeff Schneider. Spectral learning of Hidden Markov Models from dynamic and static data. In *Proceedings of 30th International Conference on Machine Learning (ICML)*, Atlanta, GA, USA, June 2013.

[74] Tâm Huynh. *Human activity recognition with wearable sensors*. PhD thesis, TU Darmstadt, September 2008.

[75] Tâm Huynh and Bernt Schiele. Analyzing features for activity recognition. In *Proceedings of Joint Conference on Smart Objects and Ambient Intelligence (sOc-EuSAI)*, pages 159–163, Grenoble, France, October 2005.

[76] Tâm Huynh and Bernt Schiele. Towards less supervision in activity recognition from wearable sensors. In *Proceedings of IEEE 10th International Symposium on Wearable Computers (ISWC)*, pages 3–10, Montreux, Switzerland, October 2006.

[77] Tâm Huynh and Bernt Schiele. Unsupervised discovery of structure in activity data using multiple eigenspaces. In *Proceedings of 2nd International Workshop on Location- and Context-Awareness (LoCA)*, Dublin, Ireland, May 2006.

[78] Tâm Huynh, Ulf Blanke, and Bernt Schiele. Scalable recognition of daily activities with wearable sensors. In *Proceedings of 3rd International Workshop on Location- and Context-Awareness (LoCA)*, pages 50–67, Oberpfaffenhofen, Germany, September 2007.

[79] Tâm Huynh, Mario Fritz, and Bernt Schiele. Discovery of activity patterns using topic models. In *Proceedings of 10th International Conference on Ubiquitous Computing (UbiComp)*, pages 10–19, Seoul, South Korea, September 2008.

[80] Stephen S. Intille, Kent Larson, Emmanuel Munguia Tapia, Jennifer S. Beaudin, Pallavi Kaushik, Jason Nawyn, and Randy Rockinson. Using a live-in laboratory for ubiquitous computing research. In *Proceedings of 4th International Conference on Pervasive Computing (PERVASIVE)*, pages 349–365, Dublin, Ireland, May 2006.

[81] Xiaobo Jin, Xinwen Hou, and Cheng-Lin Liu. Multi-class AdaBoost with hypothesis margin. In *Proceedings of 20th International Conference on Pattern Recognition (ICPR)*, pages 65–68, Washington, DC, USA, August 2010.

[82] Darcy L. Johannsen, Miguel Andres Calabro, Jeanne Stewart, Warren Franke, Jennifer C. Rood, and Gregory J. Welk. Accuracy of armband monitors for measuring daily energy expenditure in healthy adults. *Medicine and Science in Sports and Exercise*, 42(11):2134–2140, November 2010.

[83] Dean M. Karantonis, Michael R. Narayanan, Merryn Mathie, Nigel H. Lovell, and Branko G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):156–167, January 2006.

[84] Sidney Katz, Amasa B. Ford, Roland W. Moskowitz, Beverly A. Jackson, and Marjorie W. Jae. Studies of illness in the aged. The index of ADL: a standardized measure of biological and psychosocial function. *JAMA*, 185:914–919, September 1963.

[85] Ludmila I. Kuncheva. *Combining pattern classifiers: methods and algorithms*. Wiley-Interscience, 2004.

[86] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. In *Proceedings of 4th International Workshop on Knowledge Discovery from Sensor Data (SensorKDD)*, pages 74–82, Washington, DC, USA, July 2010.

[87] Cassim Ladha, Nils Y. Hammerla, Patrick Olivier, and Thomas Plötz. ClimbAX: skill assessment for climbing enthusiasts. In *Proceedings of 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pages 235–244, Zurich, Switzerland, September 2013.

[88] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9):140–150, September 2010.

[89] Óscar D. Lara, Alfredo J. Pérez, Miguel A. Labrador, and José D. Posada. Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and Mobile Computing*, 8(5):717–729, October 2012.

[90] Sian Lun Lau, Immanuel König, Klaus David, Baback Parandian, Christine Carius-Düssel, and Martin Schultz. Supporting patient monitoring using activity recognition with a smartphone. In *Proceedings of 7th International Symposium on Wireless Communication Systems (ISWCS)*, pages 810–814, York, UK, September 2010.

[91] M. Powell Lawton and Elaine M. Brody. Assessment of older people: selfmaintaining and instrumental activities of daily living. *Gerontologist*, 9(3):179–186, 1969.

[92] Michael O. Leavitt. Physical activity guidelines for Americans, 2008.

[93] Mi-hee Lee, Jungchae Kim, Kwangsoo Kim, Inho Lee, Sun Ha Jee, and Sun Kook Yoo. Physical activity recognition using a single tri-axis accelerometer. In *Proceedings of World Congress on Engineering and Computer Science (WCECS)*, San Francisco, CA, USA, October 2009.

[94] Myong-Woo Lee, Adil Mehmood Khan, Ji-Hwan Kim, Young-Sun Cho, and Tae-Seong Kim. A single tri-axial accelerometer-based real-time personal life log system capable of activity classification and exercise information generation. In *Proceedings of 32nd Annual International IEEE EMBS Conference*, pages 1390–1393, Buenos Aires, Argentina, August-September 2010.

[95] Qiang Li, John A. Stankovic, Mark A. Hanson, Adam T. Barth, John Lach, and Gang Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *Proceedings of 6th International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pages 138–143, Berkeley, CA, USA, June 2009.

[96] Lin Liao, Dieter Fox, and Henry Kautz. Extracting places and activities from GPS traces using hierarchical conditional random fields. *International Journal of Robotics Research*, 26(1):119–134, January 2007.

[97] James J. Lin, Lena Mamykina, Silvia Lindtner, Gregory Delajoux, and Henry B. Strub. Fish'n'Steps: encouraging physical activity with an interactive computer game. In *Proceedings of 8th International Conference on Ubiquitous Computing (UbiComp)*, pages 261–278, Orange County, CA, USA, September 2006.

[98] Shaopeng Liu, Robert X. Gao, Dinesh John, John Staudenmayer, and Patty S. Freedson. SVM-based multi-sensor fusion for free-living physical activity assessment. In *Proceedings of 33rd Annual International IEEE EMBS Conference*, pages 3188–3191, Boston, MA, USA, August-September 2011.

[99] Jeffrey W. Lockhart, Tony Pulickal, and Gary M. Weiss. Applications of mobile activity recognition. In *Proceedings of 14th International Conference on Ubiquitous Computing (UbiComp)*, pages 1054–1058, Pittsburgh, PA, USA, September 2012.

[100] Xi Long, Bin Yin, and Ronald M. Aarts. Single-accelerometer based daily physical activity classification. In *Proceedings of 31st Annual International IEEE EMBS Conference*, pages 6107–6110, Minneapolis, MN, USA, September 2009.

[101] Paul Lukowicz, Jamie A. Ward, Holger Junker, Mathias Stäger, Gerhard Tröster, Amin Atrash, and Thad E. Starner. Recognizing workshop activity using body worn microphones and accelerometers. In *Proceedings of 2nd International Conference on Pervasive Computing (PERVASIVE)*, pages 18–32, Linz/Vienna, Austria, April 2004.

[102] Paul Lukowicz, Andreas Timm-Giel, Michael Lawo, and Otthein Herzog. WearIT@work: toward real-world industrial wearable computing. *IEEE Pervasive Computing*, 6(4):8–13, October 2007.

[103] Paul Lukowicz, Gerald Pirkl, David Bannach, Florian Wagner, Alberto Calatroni, Kilian Förster, Thomas Holleczek, Mirco Rossi, Danial Roggen, Gerhard Tröster, Jakob Doppler, Clemens Holzmann, Andreas Riener, Alois Ferscha, and Ricardo Chavarriaga. Recording a complex, multi modal activity data set for context recognition. In *Proceedings of 23rd International Conference on Architecture of Computing Systems (ARCS), 1st Workshop on Context-Systems Design, Evaluation and Optimisation (CosDEO)*, Hannover, Germany, February 2010.

[104] Mitja Luštrek and Boštjan Kaluža. Fall detection and activity recognition with machine learning. *Informatica*, 33(2):205–212, 2009.

[105] Takuya Maekawa and Shinji Watanabe. Unsupervised activity recognition with user's physical characteristics data. In *Proceedings of IEEE 15th International Symposium on Wearable Computers (ISWC)*, pages 89–96, San Francisco, CA, USA, June 2011.

[106] Takuya Maekawa, Yutaka Yanagisawa, Yasue Kishino, Katsuhiko Ishiguro, Koji Kamei, Yasushi Sakurai, and Takeshi Okadome. Object-based activity recognition with heterogeneous sensors on wrist. In *Proceedings of 8th International Conference on Pervasive Computing (PERVASIVE)*, pages 246–264, Helsinki, Finland, May 2010.

[107] Takuya Maekawa, Yasue Kishino, Yutaka Yanagisawa, and Yasushi Sakurai. Mimic sensors: battery-shaped sensor node for detecting electrical events of handheld devices. In *Proceedings of 10th International Conference on Pervasive Computing (PERVASIVE)*, pages 20–38, Newcastle, UK, June 2012.

[108] Dominic Maguire and Richard Frisby. Comparison of feature classification algorithm for activity recognition based on accelerometer and heart rate data. In *Proceedings of 9th IT & T Conference*, Dublin, Ireland, October 2009.

[109] Jussi Mattila, Hang Ding, and Elina Mattila. Mobile tools for home-based cardiac rehabilitation based on heart rate and movement activity analysis. In *Proceedings of 31st Annual International IEEE EMBS Conference*, pages 6448–6452, Minneapolis, MN, USA, September 2009.

[110] David Mease and Abraham Wyner. Evidence contrary to the statistical view of boosting. *Journal of Machine Learning Research*, 9:131–156, June 2008.

[111] David Minnen, Tracy Westeyn, Daniel Ashbrook, Peter Presti, and Thad E. Starner. Recognizing soldier activities in the field. In *Proceedings of 4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pages 236–241, Aachen, Germany, March 2007.

[112] Lingfei Mo, Shaopeng Liu, Robert X. Gao, Dinesh John, John Staudenmayer, and Patty S. Freedson. ZigBee-based wireless multi-sensor system for physical activity assessment. In *Proceedings of 33rd Annual International IEEE EMBS Conference*, pages 846–849, Boston, MA, USA, August-September 2011.

[113] Miriam E. Nelson, W. Jack Rejeski, Steven N. Blair, Pamela W. Duncan, James O. Judge, Abby C. King, Carol A. Macera, and Carmen Castaneda-Sceppa. Physical activity and public health in older adults: recommendation from the American College of Sports Medicine and the American Heart Association. *Circulation*, 116(9):1094–1105, 2007.

[114] ROVING Networks. Roving Networks development and product website, 2013-09-16. URL `http://www.rovingnetworks.com`.

[115] Georg Ogris, Thomas Stiefmeier, Paul Lukowicz, and Gerhard Tröster. Using a complex multi-modal on-body sensor system for activity spotting. In *Proceedings of IEEE 12th International Symposium on Wearable Computers (ISWC)*, pages 55–62, Pittsburgh, PA, USA, September-October 2008.

[116] PAMAP. PAMAP project website, 2013-09-16. URL `http://www.pamap.org`.

[117] Juha Pärkkä, Miikka Ermes, Panu Korpipää, Jani Mäntyjärvi, Johannes Peltola, and Ilkka Korhonen. Activity classification using realistic data from wearable sensors. *IEEE Transactions on Information Technology in Biomedicine*, 10(1): 119–128, January 2006.

[118] Juha Pärkkä, Miikka Ermes, K. Antila, Mark van Gils, A. Mänttäri, and H. Nieminen. Estimating intensity of physical activity: a comparison of wearable accelerometer and gyro sensors and 3 sensor locations. In *Proceedings of 29th Annual International IEEE EMBS Conference*, pages 1511–1514, Lyon, France, August 2007.

[119] Juha Pärkkä, Juho Merilahti, Elina M. Mattila, Esko Malm, Kari Antila, Martti T. Tuomisto, Ari Viljam Saarinen, Mark van Gils, and Ilkka Korhonen. Relationship of psychological and physiological variables in long-term self-monitored data during work ability rehabilitation program. *IEEE Transactions on Information Technology in Biomedicine*, 13(2):141–151, March 2009.

[120] Juha Pärkkä, Luc Cluitmans, and Miikka Ermes. Personalization algorithm for real-time activity recognition using PDA, wireless motion bands, and binary decision tree. *IEEE Transactions on Information Technology in Biomedicine*, 14(5):1211–1215, September 2010.

[121] Kurt Partridge and Bo Begole. Activity-based advertising: techniques and challenges. In *Proceedings of 1st Workshop on Pervasive Advertising*, Nara, Japan, May 2009.

[122] Shyamal Patel, Chiara Mancinelli, Jennifer Healey, Marilyn Moy, and Paolo Bonato. Using wearable sensors to monitor physical activities of patients with COPD: a comparison of classifier performance. In *Proceedings of 6th International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pages 234–239, Berkeley, CA, USA, June 2009.

[123] Matthai Philipose, Kenneth P. Fishkin, Mike Perkowitz, Donald J. Patterson, Dieter Fox, Henry Kautz, and Dirk Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing*, 3(4):50–57, October 2004.

[124] Susanna Pirttikangas, Kaori Fujinami, and Tatsuo Nakajima. Feature selection and activity recognition from wearable sensors. In *Proceedings of 3rd International Symposium on Ubiquitous Computing Systems (UCS)*, pages 516–527. Seoul, South Korea, October 2006.

[125] Ronald Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, June 2010.

[126] J. Ross Quinlan. *C4.5: programs for machine learning*. San Mateo: Morgan Kaufmann, 1993.

[127] J. Ross Quinlan. Bagging, boosting and C4.5. In *Proceedings of 13th National Conference on Artificial Intelligence (AAAI)*, pages 725–730, Portland, OR, USA, August 1996.

[128] J. Ross Quinlan, Paul J. Compton, K. A. Horn, and Leslie Lazarus. Inductive knowledge acquisition: a case study. In *Proceedings of 2nd Australian Conference on Applications of Expert Systems*, pages 137–156, Sydney, Australia, May 1986.

[129] Thanawin Rakthanmanon and Eamonn J. Keogh. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *SIAM Conference on Data Mining (SDM)*, Austin, TX, USA, May 2013.

[130] Thanawin Rakthanmanon, Eamonn J. Keogh, Stefano Lonardi, and Scott Evans. MDL-based time series clustering. *Knowledge and Information Systems*, 33(2): 371–399, November 2012.

[131] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *Proceedings of 17th Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pages 1541–1546, Pittsburgh, PA, USA, July 2005.

[132] Attila Reiss. PAMAP2 Physical Activity Monitoring Data Set, 2013-09-16. URL `http://archive.ics.uci.edu/ml/datasets/PAMAP2+Physical+Activity+Monitoring`.

[133] Attila Reiss and Didier Stricker. Towards global aerobic activity monitoring. In *Proceedings of 4th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, Crete, Greece, May 2011.

[134] Attila Reiss and Didier Stricker. Introducing a modular activity monitoring system. In *Proceedings of 33rd Annual International IEEE EMBS Conference*, pages 5621–5624, Boston, MA, USA, August-September 2011.

[135] Attila Reiss and Didier Stricker. Creating and benchmarking a new dataset for physical activity monitoring. In *Proceedings of 5th Workshop on Affect and Behaviour Related Assistance (ABRA)*, Crete, Greece, June 2012.

[136] Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *Proceedings of IEEE 16th International Symposium on Wearable Computers (ISWC)*, pages 108–109, Newcastle, UK, June 2012.

[137] Attila Reiss and Didier Stricker. Aerobic activity monitoring: towards a long-term approach. *International Journal of Universal Access in the Information Society (UAIS)*, March 2013.

[138] Attila Reiss and Didier Stricker. Personalized mobile physical activity recognition. In *Proceedings of IEEE 17th International Symposium on Wearable Computers (ISWC)*, Zurich, Switzerland, September 2013.

[139] Attila Reiss, Markus Weber, and Didier Stricker. Exploring and extending the boundaries of physical activity recognition. In *Proceedings of 2011 IEEE International Conference on Systems, Man and Cybernetics (SMC), Workshop on*

Robust Machine Learning Techniques for Human Activity Recognition, pages 46–50, Anchorage, AK, USA, October 2011.

[140] Attila Reiss, Ilias Lamprinos, and Didier Stricker. An integrated mobile system for long-term aerobic activity monitoring and support in daily life. In *Proceedings of 2012 International Symposium on Advances in Ubiquitous Computing and Networking (AUCN)*, Liverpool, UK, June 2012.

[141] Attila Reiss, Gustaf Hendeby, and Didier Stricker. A competitive approach for human activity recognition on smartphones. In *Proceedings of 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Bruges, Belgium, April 2013.

[142] Attila Reiss, Gustaf Hendeby, and Didier Stricker. Towards robust activity recognition for everyday life: methods and evaluation. In *Proceedings of 7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, Venice, Italy, May 2013.

[143] Attila Reiss, Gustaf Hendeby, and Didier Stricker. Confidence-based multiclass AdaBoost for physical activity monitoring. In *Proceedings of IEEE 17th International Symposium on Wearable Computers (ISWC)*, Zurich, Switzerland, September 2013.

[144] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jacob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creature, and José del R. Millán. Collecting complex activity datasets in highly rich networked sensor environments. In *Proceedings of 7th International Conference on Networked Sensing Systems (INSS)*, pages 233–240, Kassel, Germany, June 2010.

[145] Daniel Roggen, Stephane Magnenat, Markus Waibel, and Gerhard Tröster. Wearable computing: designing and sharing activity recognition systems across platforms. *IEEE Robotics and Automation Magazine*, 18(2):83–95, June 2011.

[146] Mirco Rossi, Gerhard Tröster, and Oliver Amft. Recognizing daily life context using web-collected audio data. In *Proceedings of IEEE 16th International Symposium on Wearable Computers (ISWC)*, pages 25–28, Newcastle, UK, June 2012.

[147] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: a modern approach*. Prentice Hall, Englewood Cliffs, 2010.

[148] Maytal Saar-Tsechansky and Foster Provost. Handling missing values when applying classification models. *Journal of Machine Learning Research*, 8:1625–1657, December 2007.

[149] Hesam Sagha, Sundara Tejaswi Digumarti, Ricardo Chavarriaga, Alberto Calatroni, Daniel Roggen, and Gerhard Tröster. Benchmarking classification techniques using the Opportunity human activity dataset. In *Proceedings of 2011 IEEE International Conference on Systems, Man and Cybernetics (SMC), Workshop on Robust Machine Learning Techniques for Human Activity Recognition*, pages 36–40, Anchorage, AK, USA, October 2011.

[150] Alireza Sahami Shirazi, James Clawson, Yashar Hassanpour, Mohammad J. Tourian, Albrecht Schmidt, Ed H. Chi, Marko Borazio, and Kristof van Laerhoven. Already up? using mobile phones to track & share sleep behavior. *International Journal of Human-Computer Studies*, 71(9):878–888, September 2013.

[151] Ralf Salomon, Marian Lüder, and Gerald Bieber. iFall - a new embedded system for the detection of unexpected falls. In *Proceedings of 8th IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 286–291, Mannheim, Germany, March-April 2010.

[152] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2): 197–227, June 1990.

[153] Robert E. Schapire. Using output codes to boost multiclass learning problems. In *Proceedings of 14th International Conference on Machine Learning (ICML)*, pages 313–321, Nashville, TN, USA, July 1997.

[154] Robert E. Schapire. Computer science 511, foundations of machine learning, lecture 14. Department of Computer Science, Princeton University, 2006.

[155] Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, December 1999.

[156] Bill N. Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Proceedings of First Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 85–90, Santa Cruz, CA, USA, December 1994.

[157] Markus Scholz, Stephan Sigg, Gerrit Bagschik, Toni Guenther, Georg von Zengen, Dimana Shishkova, Yusheng Ji, and Michael Beigl. SenseWaves: radiowaves for context recognition. In *Proceedings of 9th International Conference on Pervasive Computing (PERVASIVE)*, San Francisco, CA, USA, June 2011.

[158] Markus Scholz, Stephan Sigg, Hedda R. Schmidtke, and Michael Beigl. Challenges for device-free radio-based activity recognition. In *Proceedings of 8th International ICST Conference on Mobile and Ubiquitous Systems (MobiQuitous), 3rd Workshop on Context-Systems Design, Evaluation and Optimisation (CosDEO)*, Copenhagen, Denmark, December 2011.

[159] Burr Settles. Active learning literature survey. Technical Report 1648, Computer Sciences, University of Wisconsic-Madison, 2009.

[160] SHIMMER. Shimmer development and product website, 2013-09-16. URL `http://www.shimmersensing.com`.

[161] Hua Si, Seung Jin Kim, Nao Kawanishi, and Hiroyuki Morikawa. A context-aware reminding system for daily activities of dementia patients. In *Proceedings of 27th International Conference on Distributed Computing Systems Workshops (ICDCS)*, Toronto, ON, Canada, June 2007.

[162] J. Paul Siebert. *Vehicle recognition using rule based methods*. Turing Institute (Glasgow, Scotland), 1987.

[163] SONY. Sony SmartWatch product website, 2013-09-16. URL `http://www.sonymobile.com/gb/products/accessories/smartwatch`.

[164] Claudio De Stefano, Antonio Della Cioppa, and Angelo Marcelli. An adaptive weighted majority vote rule for combining multiple classifiers. In *Proceedings of 16th International Conference on Pattern Recognition (ICPR)*, pages 192–195, Quebec, QC, Canada, August 2002.

[165] Thomas Stiefmeier, Daniel Roggen, Georg Ogris, Paul Lukowicz, and Gerhard Tröster. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing*, 7(2):42–50, April 2008.

[166] Maja Stikic, Tâm Huynh, Kristof van Laerhoven, and Bernt Schiele. ADL recognition based on the combination of RFID and accelerometer sensing. In *Proceedings of 2nd International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pages 258–263, Tampere, Finland, January-February 2008.

[167] Johannes A. Stork, Luciano Spinello, Jens Silva, and Kai O. Arras. Audio-based human activity recognition using non-Markovian ensemble voting. In *Proceedings of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 509–514, Paris, France, September 2012.

[168] Christina Strohrmann, Holger Harms, and Gerhard Tröster. What do sensors know about your running performance? In *Proceedings of IEEE 15th International Symposium on Wearable Computers (ISWC)*, pages 101–104, San Francisco, CA, USA, June 2011.

[169] Amarnag Subramanya, Alvin Raj, Je Bilmes, and Dieter Fox. Recognizing activities and spatial context using wearable sensors. In *Proceedings of 22nd Conference on Uncertainty in Artificial Intelligence (UAI)*, Cambridge, MA, USA, July 2006.

[170] Feng-Tso Sun, Cynthia Kuo, Heng-Tze Cheng, Senaka Buthpitiya, Patricia Collins, and Martin L. Griss. Activity-aware mental stress detection using physiological signals. In *Proceedings of 2nd International ICST Conference on Mobile Computing, Applications, and Services (MobiCASE)*, pages 211–230, Santa Clara, CA, USA, October 2010.

[171] Xu Sun, Hisashi Kashima, Ryota Tomioka, Naonori Ueda, and Ping Li. A new multi-task learning method for personalized activity recognition. In *Proceedings of IEEE 11th International Conference on Data Mining (ICDM)*, pages 1218–1223, Vancouver, BC, Canada, December 2011.

[172] Sudeep Sundaram and Walterio W. Mayol Cuevas. High level activity recognition using low resolution wearable vision. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Workshop on Egocentric Vision*, pages 25–32, Miami, FL, USA, June 2009.

[173] Emmanuel Munguia Tapia, Stephen S. Intille, William Haskell, Kent Larson, Julie Wright, Abby King, and Robert Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *Proceedings of IEEE 11th International Symposium on Wearable Computers (ISWC)*, pages 1–4, Boston, MA, USA, October 2007.

[174] Moritz Tenorth, Jan Bandouch, and Michael Beetz. The TUM Kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *Proceedings of IEEE 12th International Conference on Computer Vision (ICCV), Workshop on Tracking Humans for the Evaluation of Their Motion in Image Sequences (THEMIS)*, Kyoto, Japan, September-October 2009.

[175] Bruce H. Thomas. Have we achieved the ultimate wearable computer? In *Proceedings of IEEE 16th International Symposium on Wearable Computers (ISWC)*, pages 104–107, Newcastle, UK, June 2012.

[176] TinyOS. TinyOS home page, 2013-09-16. URL `http://www.tinyos.net`.

[177] Dorra Trabelsi, Samer Mohammed, Faicel Chamroukhi, Latifa Oukhellou, and Yacine Amirat. Supervised and unsupervised classification approaches for human activity recognition using body-mounted sensors. In *Proceedings of 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 417–422, Bruges, Belgium, April 2012.

[178] TRIVISIO. Trivisio Prototyping GmbH development and product website, 2013-09-16. URL `http://www.trivisio.com`.

[179] Wallace Ugulino, Debora Cardador, Katia Vega, Eduardo Velloso, Ruy Milidiú, and Hugo Fuks. Wearable computing: accelerometers' data classification of body postures and movements. In *Proceedings of 21st Brazilian Symposium on Artificial Intelligence (SBIA)*, pages 52–61, Curitiba, Brazil, October 2012.

[180] Tim L.M. van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of 10th International Conference on Ubiquitous Computing (UbiComp)*, pages 1–9, Seoul, South Korea, September 2008.

[181] Tim L.M. van Kasteren, Hande Alemdar, and Cem Ersoy. Effective performance metrics for evaluating activity recognition methods. In *Proceedings of 24th International Conference on Architecture of Computing Systems (ARCS)*, Como, Italy, February 2011.

[182] Tim L.M. van Kasteren, Gwenn Englebienne, and Ben Kröse. Hierarchical activity recognition using automatically clustered actions. In *Proceedings of 2nd International Conference on Ambient Intelligence (AmI)*, pages 82–91, Amsterdam, Netherlands, November 2011.

[183] Kristof van Laerhoven, Marko Borazio, David Kilian, and Bernt Schiele. Sustained logging and discrimination of sleep postures with low-level, wrist-worn sensors. In *Proceedings of IEEE 12th International Symposium on Wearable Computers (ISWC)*, pages 69–76, Pittsburgh, PA, USA, September-October 2008.

[184] Alexander Vezhnevets and Vladimir Vezhnevets. Modest AdaBoost - teaching AdaBoost to generalize better. In *Proceedings of 15th International Conference on Computer Graphics and Applications (Graphicon)*, Novosibirsk, Russia, June 2005.

[185] VILIV. Viliv S5 product website, 2013-09-16. URL `http://www.myviliv.com/ces/main_s5.html`.

[186] Elena Villalba, Manuel Ottaviano, María Teresa Arredondo, A. Martinez, and S. Guillen. Wearable monitoring system for heart failure assessment in a mobile environment. *Computers in Cardiology*, 33:237–240, 2006.

[187] Matteo Voleno, Stephen J. Redmond, Sergio Cerutti, and Nigel H. Lovell. Energy expenditure estimation using triaxial accelerometry and barometric pressure measurement. In *Proceedings of 32nd Annual International IEEE EMBS Conference*, pages 5185–5188, Buenos Aires, Argentina, August-September 2010.

[188] Jamie A. Ward, Paul Lukowicz, Gerhard Tröster, and Thad E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10): 1553–1567, October 2006.

[189] Jamie A. Ward, Paul Lukowicz, and Hans W. Gellersen. Performance metrics for activity recognition. *ACM Transactions on Intelligent Systems and Technology*, 2(1), January 2011. Article No. 6.

[190] Gary M. Weiss and Jeffrey W. Lockhart. The impact of personalization on smartphone-based activity recognition. In *Proceedings of 26th Conference on Artificial Intelligence (AAAI), Workshop on Activity Context Representation*, Toronto, ON, Canada, July 2012.

[191] WHO. World Health Organization (WHO) fact sheet No.311 (obesity and overweight), 2013-09-16. URL `http://www.who.int/mediacentre/factsheets/fs311`.

[192] Jacqueline Wijsman, Bernard Grundlehner, Hao Liu, Hermie Hermens, and Julien Penders. Towards mental stress detection using wearable physiological sensors. In *Proceedings of 33rd Annual International IEEE EMBS Conference*, pages 1798–1801, Boston, MA, USA, August-September 2011.

[193] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: practical machine learning tools and techniques*. Morgan Kaufmann, Burlington, 2011.

[194] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, December 2007.

[195] Yang Xue and Lianwen Jin. A naturalistic 3D acceleration-based activity dataset & benchmark evaluations. In *Proceedings of 2010 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 4081–4085, Istanbul, Turkey, October 2010.

[196] Zhixian Yan, Vigneshwaran Subbaraju, Dipanjan Chakraborty, Archan Misra, and Karl Aberer. Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach. In *Proceedings of IEEE 16th International Symposium on Wearable Computers (ISWC)*, pages 17–24, Newcastle, UK, June 2012.

[197] Koji Yatani and Khai N. Truong. BodyScope: a wearable acoustic sensor for activity recognition. In *Proceedings of 14th International Conference on Ubiquitous Computing (UbiComp)*, pages 341–350, Pittsburgh, PA, USA, September 2012.

[198] Wojtek Zajdel, Johannes D. Krijnders, Tjeerd Andringa, and Dariu M. Gavrila. CASSANDRA: audio-video sensor fusion for aggression detection. In *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 200–205, London, UK, September 2007.

[199] ZEPHYR. Zephyr Technology development and product website, 2013-09-16. URL `http://www.zephyranywhere.com`.

[200] Yi Zhan and Tadahiro Kuroda. Wearable sensor-based human activity recognition from environmental background sounds. *Journal of Ambient Intelligence and Humanized Computing*, May 2012.

[201] Zhongtang Zhao, Yiqiang Chen, Junfa Liu, Zhiqi Shen, and Mingjie Liu. Cross-people mobile-phone based activity recognition. In *Proceedings of 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2545–2550, Barcelona, Spain, July 2011.

[202] Ji Zhu, Saharon Rosset, Hui Zou, and Trevor Hastie. Multi-class adaboost. Technical Report 430, Department of Statistics, University of Michigan, 2005.

[203] Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. Multi-class Adaboost. *Statistics and Its Interface*, 2:349–360, 2009.

[204] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsic-Madison, 2008.

[205] Andreas Zinnen, Ulf Blanke, and Bernt Schiele. An analysis of sensor-oriented vs. model-based activity recognition. In *Proceedings of IEEE 13th International Symposium on Wearable Computers (ISWC)*, pages 93–100, Linz, Austria, September 2009.

Curriculum Vitæ

Name: Attila Reiss
Website: http://sites.google.com/site/attilareiss/

Education

Jan 2014 Doctor of Engineering, Department of Computer Science,
Technical University of Kaiserslautern
Thesis: *Personalized Mobile Physical Activity Monitoring for Everyday Life*

Jan 2008 M.Sc. in Electrical Engineering, Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics
Minor in *Embedded Information Systems* and *Intelligent Systems*
Thesis: *Formale Beschreibungssprache von Parametern für Geräte zum Bedienen und Beobachten in der Automatisierungstechnik*

Fall 2005 Semester abroad, Faculty of Electrical Engineering and Information Technology,
Vienna University of Technology

June 2002 Bányai Júlia Grammar School, Kecskemét, Hungary

Professional Experience

Feb 2009 - Dec 2013 Researcher, Department of Augmented Vision,
German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany

Feb 2009 - Jan 2011 Software engineer (50%), **Rittal GmbH & Co. KG**, Herborn, Germany

Oct 2007 - Jan 2009 Software developer, **evosoft GmbH (Siemens AG Automation and Drives)**, Fürth, Germany

Attila Reiss
Kaiserslautern, 10 January 2014.