Sensor-based Human Activity Recognition: Overcoming Issues in a Real World Setting

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von
Timo Sztyler

Mannheim, 2019

Dekan:	Dr. Bernd Lübcke, Universität Mannheim
Referent:	Prof. Dr. Heiner Stuckenschmidt, Universität Mannheim
Korreferent:	Prof. Dr. Daniele Riboni, University of Cagliari, Italien

Tag der mündlichen Prüfung: 09.05.2019

Acknowledgments

The first person I would like to thank is my supervisor Heiner Stuckenschmidt: he always gave me the freedom to evolve and to make my own decisions. His wise guidance and support were essential for this work. Special thanks to Gabriele Civitarese: collaborating and discussing with you on challenging problems was always great! Thanks to Daniele Ribioni and Claudio Bettini for fruitful discussions and collaborations. Thanks to my parents Christiane Sztyler and Andrzej Sztyler, which never stopped believing in me. Thanks to my friend, Jakob Huber who has always been there for me to share laughs and tears. Thanks to all the friends who have been at my side despite everything. I also want to thank Christian Meilicke, Jan Noessner, and Mathias Niepert for their recommendation and advice whether I should try to take a PhD degree. Finally, I also want to thank all my (other) co-authors, each and every one of you taught me something (in alphabetical order): Christian Becker, Martin Breitbach, Josep Carmona, Alexander Diete, Janick Edinger, Christian Krupitzer, Wolfgang Petrich, Johanna Völker, and Lydia Weiland.

Abstract

The rapid growing of the population age in industrialized societies calls for advanced tools to continuous monitor the activities of people. The goals of those tools are usually to support active and healthy ageing, and to early detect possible health issues to enable a long and independent life. Recent advancements in sensor miniaturization and wireless communications have paved the way to unobtrusive activity recognition systems. Hence, many pervasive health care systems have been proposed which monitor activities through unobtrusive sensors and by machine learning or artificial intelligence methods. Unfortunately, while those systems are effective in controlled environments, their actual effectiveness out of the lab is still limited due to different shortcomings of existing approaches.

In this work, we explore such systems and aim to overcome existing limitations and shortcomings. Focusing on physical movements and crucial activities, our goal is to develop robust activity recognition methods based on external and wearable sensors that generate high quality results in a real world setting. Under laboratory conditions, existing research already showed that wearable sensors are suitable to recognize physical activities while external sensors are promising for activities that are more complex. Consequently, we investigate problems that emerge when coming out of the lab. This includes the position handling of wearable devices, the need of large expensive labeled datasets, the requirement to recognize activities in almost real-time, the necessity to adapt deployed systems online to changes in behavior of the user, the variability of executing an activity, and to use data and models across people. As a result, we present feasible solutions for these problems and provide useful insights for implementing corresponding techniques. Further, we introduce approaches and novel methods for both external and wearable sensors where we also clarify limitations and capabilities of the respective sensor types. Thus, we investigate both types separately to clarify their contribution and application use in respect of recognizing different types of activities in a real world scenario.

Overall, our comprehensive experiments and discussions show on the one hand the feasibility of physical activity recognition but also recognizing complex activities in a real world scenario. Comparing our techniques and results with existing works and state-of-the-art techniques also provides evidence concerning the reliability and quality of the proposed techniques. On the other hand, we also identify promising research directions and highlight that combining external and wearable sensors seem to be the next step to go beyond activity recognition. In other words, our results and discussions also show that combining external and wearable sensors would compensate weaknesses of the individual sensors in respect of certain activity types and scenarios. Therefore, by addressing the outlined problems, we pave the way for a hybrid approach. Along with our presented solutions, we conclude our work with a high-level multi-tier activity recognition architecture showing that aspects like physical activity, (emotional) condition, used objects, and environmental features are critical for reliable recognizing complex activities.

Zusammenfassung

Das rasante Wachstum der älteren Bevölkerung in den Industriegesellschaften ruft nach fortschrittlichen Lösungen zur kontinuierlichen Erkennung alltäglicher Aktivitäten. Dies soll die Unterstützung des aktiven und gesunden Alterns und die frühzeitige Erkennung möglicher Gesundheitsprobleme ermöglichen und so ein längeres und unabhängiges Leben fördern. Die jüngsten Fortschritte in der Miniaturisierung von Sensoren und der drahtlosen Kommunikation haben den Weg für diese Art von Aktivitätserkennungssysteme geebnet. Existierende Ansätze sind allerdings nur in kontrollierter Umgebung wirksam und in realer Umgebung häufig nicht erprobt und aufgrund von Mängeln limitiert.

In dieser Arbeit untersuchen wir solche Systeme, mit der Absicht bestehende Limitierungen und Mängel zu überwinden. Unser Ziel ist es, zuverlässige Methoden zur Aktivitätserkennung zu entwickeln, die auf externen und tragbaren Sensoren basieren und zudem qualitativ hochwertige Ergebnisse in einem realen Szenario liefern. Forschungen haben bereits gezeigt, dass unter Laborbedingungen tragbare Sensoren für die Erkennung von körperliche Bewegungen und externe Sensoren für die Erkennung von komplexere Aktivitäten geeignet sind. In diesem Zusammenhang, untersuchen wir Probleme die auftreten, sobald man versucht Aktivitäten unter realen Bedingungen zu erkennen. Dazu gehört die variierende Position von tragbaren Geräten, die Hürde hinsichtlich benötigter, umfangreicher und annotierter Datensätze, die Anforderung die Aktivitäten in Echtzeit zu erkennen, die Möglichkeit laufende Systeme online an Änderungen des Nutzerverhaltens anzupassen, die Vielfältigkeit mit der eine Aktivität ausgeführt werden kann und nicht zuletzt die personenübergreifende Nutzung von Daten und Modellen. Als Ergebnis präsentieren wir praktikable Lösungen und Erkenntnisse für diese Probleme. Damit einhergehend, stellen wir Ansätze und neuartige Methoden für externe und tragbare Sensoren vor, wobei wir auch die Grenzen und Möglichkeiten der jeweiligen Sensortypen verdeutlichen. Folglich untersuchen wir beide Arten getrennt, um ihren Beitrag und ihre Verwendung in einem realen Szenario zu klären.

Unsere umfangreichen Experimente und Diskussionen zeigen die Machbarkeit der Erkennung von körperlichen und komplexen Aktivitäten unter realen Bedingungen. Darüber hinaus unterstreicht der Vergleich mit bestehenden Forschungsarbeiten, die Zuverlässigkeit und Qualität der vorgeschlagenen Lösungen. Auf der anderen Seite identifizieren wir weitere und vielversprechende Forschungsrichtungen und erkennen, dass die Kombination von externen und tragbaren Sensoren der nächste logische Schritt zu sein scheint. Unsere Ergebnisse und Diskussionen zeigen, dass die Kombination dieser Sensortypen Schwächen des jeweils anderen in Bezug auf bestimmte Aktivitätsarten aber auch Szenarien kompensieren würde. Folglich, ebnen wir den Weg für einen hybriden Ansatz durch die Lösung der aufgezeigten Probleme. Zusammen mit unseren vorgestellten Ergebnissen schließen wir unsere Arbeit mit einer konzeptuellen Architektur zur Aktivitätserkennung ab. Diese zeigt, dass Aspekte wie körperliche Aktivität, (emotionaler) Zustand, verwendete Objekte und Umgebungsmerkmale wichtig für eine zuverlässige Erkennung sind.

Contents

List of Figures

List of Tables

List of Selected Publications

Parts of the work presented in this thesis have been published in international conferences and journals. For all publications, the author of this thesis was a key contributor (see Appendix A). While introducing our methods and results, we will refer to the respective publications. In the following, we list the main publications ordered by year:

- T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE Computer Society, 2016, pp. 1–9, doi: 10.1109/PERCOM.2016.7456521.
- D. Riboni, T. Sztyler, G. Civitarese, and H. Stuckenschmidt, "Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning," in Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2016, pp. 1–12, doi: 10.1145/2971648.2971691.
- T. Sztyler, H. Stuckenschmidt, and W. Petrich, "Position-aware activity recognition with wearable devices," Pervasive and Mobile Computing, vol. 38, no. Part 2, pp. 281–295, 2017, doi: 10.1016/j.pmcj.2017.01.008.
- T. Sztyler and H. Stuckenschmidt, "Online personalization of cross-subjects based activity recognition models on wearable devices," in 2017 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE Computer Society, 2017, pp. 180–189, doi: 10.1109/PERCOM.2017.7917864.
- T. Sztyler, "Towards real world activity recognition from wearable devices," in 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE Computer Society, 2017, pp. 97–98, doi: 10.1109/PERCOMW.2017.7917535.
- T. Sztyler, G. Civitarese, and H. Stuckenschmidt, "Modeling and reasoning with Problog: An application in recognizing complex activities," in 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE Computer Society, 2018, pp. 781–786, doi: 10.1109/PERCOMW.2018.8480299.
- C. Krupitzer, T. Sztyler, J. Edinger, M. Breitbach, H. Stuckenschmidt, and C. Becker, "Hips do lie! A position-aware mobile fall detection system," in 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE Computer Society, 2018, pp. 95–104, doi: 10.1109/PERCOM.2018.8444583.
- G. Civitarese, C. Bettini, T. Sztyler, D. Riboni, and H. Stuckenschmidt, "NECTAR: Knowledge-based collaborative active learning for activity recognition," in 2018 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE Computer Society, 2018, pp. 125–134, doi: 10.1109/PERCOM.2018.8444590.

Chapter 1

Introduction

The rapid growing of the population age in industrialized societies calls for advanced tools to continuous monitor the activities of people. The goals of those tools are usually to support active and healthy ageing, and to early detect possible health issues to enable a long and independent life [9, 10]. Especially dietary risks and insufficient physical activities but also the absence of needed help can lead to difficult-to-treat long-term effects. The loss of self-confidence and the change in behavior to prevent issues in everyday situations can cause a physical as well as a psychological decline in health that in turn results in a premature death [11, 12]. Recent advancements in sensor miniaturization and wireless communications have paved the way to unobtrusive activity recognition systems. Hence, many pervasive health care systems have been proposed which monitor activities through unobtrusive sensors and by machine learning or artificial intelligence methods. Knowledge about the activities carried out by individuals is a fundamental requirement [13]. Unfortunately, while those systems are effective in controlled environments, their actual effectiveness out of the lab is still limited due to different shortcomings of existing approaches.

Human Activity Recognition has been deeply investigated in the last decade taking advantage of the effective sensing infrastructure that is becoming available with off-the-shelf products as part of domotics, smart objects and wearable devices. Indeed, domains of activity-aware computing range from smart-homes and e-health, to gaming, smart manufacturing, pervasive advertising, and smart-cities. Among the many applications in mobile and pervasive computing, the continuous recognition of *Activities of Daily Living* has been identified as a key enabler of assisted living and e-health systems [9,14]. Indeed, recognizing those activities not only allows verifying if someone performed a certain activity but also enables to reconstruct in the end the daily routine of a person. Being able to recognize the daily routine allows to learn the user's behavior that in turn facilitates to optimize the course of the day in respect of food intake or sport. In addition, predefined patterns like medical instructions could be easily verified and the gained knowledge can be reused to improve the overall recognition performance. Of course, data security and the users' privacy go along with these scenarios and need to be considered in architectural design decisions.

Having said this, state-of-the-art human activity recognition systems are far from being able to achieve this. For that reason, first we want to clarify and outline the term *Activity Recognition* and the associated research directions where we focus on systems which rely on wearable or external sensors. Further, we also present open issues of existing activity recognition approaches followed by our goals, research questions and contributions.

1.1 Human Activity Recognition

Human Activity Recognition (HAR) is a general term for describing research that deals with interpreting recorded sensor data or signals to determine the activity that initially triggered them. One of the first works on HAR date back to 1999 [15] where researchers tried to detect certain motions and postures with accelerometers. Today, researchers use and investigate several different kinds of sensors including motion, proximity, environmental, video, and physiological sensors. In this context, a distinction is made whether the environment is equipped with those sensors (external sensors) or if they are attached directly to or carried by the user (wearable sensors).

External sensors are usually fixed to preselected objects or locations to recognize interactions. The essential idea is that the user has to interact with those objects or has to be present in a certain location while performing the activity of interest. Then, the recorded sequence of sensor events is used to decide which activity was performed. Intelligent- or smart-homes are typical examples of external sensing [5, 16, 17]. These systems are able to recognize fairly complex activities like taking medicine but in turn are restricted to a certain environment. We denote an activity as complex when it is characterized by the user's posture or motion and an active interaction with the environment. In contrast, wearable sensors are carried by the user and are mostly used to recognize simpler activities like motions and postures [1, 18, 19]. For that purpose, the sensors are attached to certain body parts to capture the movements of these body parts. Analyzing the recorded sensor data allows to recognize which simple activity (e.g., walking) was performed by the user. Complex activities are usually not targeted, as the body movements alone are insufficient to capture those activities [20]. Of course, there are also wearable sensors that capture the users' surrounding such as first-person video cameras (smart-glasses). These are upcoming approaches, which try to use the advantages of wearable devices but aiming to recognize complex activities. Indeed, an essential advantage of the wearable sensors is the fact that they are not bound to a certain location. Overall, external and wearable sensor based approaches can be considered as two fundamental branches of the HAR research area.

As one notice, the term *Activity* is broadly used and represents essentially different activity types including simple actions, physical activities, and complex activities like Activities of Daily Living (ADL). While these types are well established and distinguished in the domain of HAR, unfortunately there exists no agreed set of activities. Basically, the term *physical activities* covers postures and locomotions where the most frequently considered activities are walking, running, climbing stairs, standing, sitting, and lying. In contrast, less common are jumping and crawling. In respect of complex activities, researchers mostly focus on ADLs. The term ADL comes originally from the health care area and refers to people's daily self-care activities [21]. The term ADLs consist of activities like grooming, eating, dressing, and cleaning and are often divided into Basic ADLs (BADLs), Instrumental ADLs (IADLs), and Personal ADLs (PADLs) [22]. While it seems quite intuitive which kind of activities are represented by those activity types, a closer look leads to ambiguities. For instance, *showering* can be considered as a more detailed description of *grooming* but both are described as ADL [22]. Further, usually it is necessary to perform a physical activity while executing a complex one but few works associate these two activity types. Hence, the relation and hierarchy between those activities is often unclear or just not considered (e.g. sitting vs. cycling [23], cleaning vs. sweeping [24], or standing vs. brushing [25]). Indeed, it may depend on the scenario which degree of detail is required but the missing abstraction level of activities, so a common agreement, makes it hard to compare activity recognition results across different sets of activities. Similar issues arise when comparing simple actions like grabbing and continual physical activities. However, basically the existing research can be grouped by these dimensions, i.e., the targeted type of activity, the used sensor types, and its position (external vs. wearable).

As this suggests, the HAR research area is fragmented meaning that there are many approaches using different setups and focusing on different problems of the same application but do not combine them. For instance, researcher propose the recognition of physical activities as it may help diabetes patients which have often to follow a well-defined exercise routine [26]. However, similarly the recognition of ADLs is reasoned, as also complex activities like foot intake or the use of medication are important because they influence the blood sugar level. Indeed, these approaches do not exclude but would even complement each other but so far. only few works discuss or consider this idea to overcome existing limitations [24]. In the following, we go into detail and introduce both branches, i.e., external and wearable sensor based approaches, by characterizing open issues, research interests, and by clarifying for what we aim.

1.2 Problem Statement

Due to the variety of possibilities in recognizing activities, first we want to clarify what we want to achieve so which setup is suitable and which activities are essential. For that purpose, in the following, we outline our scenario focusing on supporting diabetes patients and subsequently we introduce open issues in respect of existing techniques to give the reader an idea about the state-of-the-art. In this way, one the one hand we address the mentioned issue regarding which level of detail of an activity is required (*Managing* vs. *Preparing and Taking* Medications). On the other hand, we highlight which problems we have to address to build a pervasive health care system which is able to support diabetes patients in real life situations.

1.2.1 Diabetes Mellitus

Today, more than 425 million people have *Diabetes Mellitus* [27]. It can be considered as a metabolic disorder, which is characterized by an increased blood sugar level. According to the WHO [28], a person has *Diabetes Mellitus* when on an empty stomach the blood sugar is $\geq$ 126 mg/dl or in a random point in time $\geq$ 200 mg/dl.

Usually, the digestive system of a human decomposes carbohydrates of food into glucose. Then, the glucose is absorbed into the blood so it is distributed to the body. As a counteraction, the pancreas produces the hormone insulin to regulate the sugar level of the blood. In this context, insulin enables that the body's cells are able to absorb the glucose from the blood, i.e., to store it as energy. If the insulin production is disturbed, i.e., when the pancreas does not produce as much insulin as it is required to regulate the blood sugar level then the amount of sugar increases. This entails tiredness, decreased vision, and sickness. In the long term, so when the blood sugar level stays high, this may even lead to a hypoglycemia that can cause toxic acids which goes along with confusion, abdominal pain, and coma [29]. As a side effect, the body starts to use fat and protein cells instead of the glucose as energy. In an extreme case, the accelerated destruction of fat cells causes deposits that in turn lead to an abnormal risk of a heart attack [30].

The WHO distinguishes between certain types of diabetes where *Type-1* and *Type-2* are the most common one [28]. Thus, there are different reasons that can cause the disease, which goes along with different methods of treatment. For instance, *Type-1* results from an autoimmune disease that destroys the part of the pancreas that is responsible for the insulation production. In contrast, *Type-2* is caused by an insulin resistance; hence, the insulin production works as expected but the body' cells require more insulin to absorb the glucose. At a certain point, the pancreas is unable to produce as much insulin as required by the body's cells. The latter is the most common type of diabetes.

The treatment of diabetes lasts as long as life, i.e., there is no cure but only methods which help to control the blood sugar level. This includes a balanced nutrition, physical exercises, medication, and insulin [31]. This holds especially for *Type-2* also known as *adult-onset diabetes*. Indeed, this term is considered as obsolete because today already several young people have this type of diabetes. However, this does not mean that the number of elderly diagnosed with diabetes decreased and especially this group needs support in respect of everyday situations [32].

In our work, we focus on supporting people in respect of physical exercises and crucial activities such as intake of food or medication aiming to avoid the mentioned dangers. For this purpose, we want to recognize the corresponding physical activities and ADLs to provide information whether a person has a fair amount of exercise and performs the ADLs of interest. For that reason, in the following first we outline the open issues concerning recognizing physical activities with wearable sensors and subsequently open issues of recognizing ADLs by using external sensors.

1.2.2 Activity Recognition with Wearable Sensors

During the last two decades, especially acceleration sensors were investigated for recognizing physical activities. Researchers attached them to certain body parts of the user to capture the movements of these body parts. Then, the recorded acceleration data was analyzed to determine which physical activity was probably performed. Based on this idea, several experiments were performed under laboratory conditions achieving promising results by recognizing, e.g., walking or running [33–35]. The development of wearable devices such as smart-phones, smart-watches, smart-glasses, and fitness wristbands in the last years encouraged this research and resulted in an increased focus on out of the lab experiments. Those devices feature a variety of sensors that are carried all day long by many people (compare [36]). On one the hand, this allows easily to rely on additional inertial sensors such as the gyroscope and magnetometer but also to monitor the heart rate or sweating. On the other hand, the step out of the lab resulted in several new and unaddressed problems. First, usually it is up to the user where to carry a wearable device so its position is not known a-priori and may change over time. In this context, several works clarify that different body parts produce different motion patterns for the same activity, which in turn has an influence on the activity recognition quality. Second, most existing approaches rely on machine learning techniques, i.e., the target user has to collect and label sensor data for building a classification model (in the following denoted as *single-subject*). This is often not feasible, especially in our scenario where elders or patients should be observed. Third, proposed activity recognition solutions do not take into account that the movement pattern of a person could change due to age, injuries, or a varying fitness level. This means that the performance of the recognition system will drop over time.

Of course, researchers are aware of these problems but they got too little attention and existing approaches are limited. For instance, researchers investigated the possibility to recognize the on-body device position while walking by matching prior-recorded patterns of that activity for each considered position. Hence, a change of the device's position might be not immediately detected leading to a miss-interpretation of the sensor data. Further, several researches investigated the performance of classifiers that were trained with all available data (also referred to as *leave-one-subject-out*) [23, 37] to have a classification model immediately at hand. However, these approaches performed often significantly worse than a single-subject approach. In addition, such an approach does not scale in respect of large or in-homogeneous groups of people. For instance, children walk in a different way as elderly but also the body type is an influencing factor. This implies that actually a group-based approach is required. Moreover, researchers also investigated the concept of co-training [38] and parameter adaptation [39] for personalization a classification models. Actually, such an adaption is of general use as it can be used to increase the performance of a *leave-one-subject-out* based model but also to adapt it over time. The drawback is that the proposed approaches require re-training and to save the training data permanently. Thus, there are attempts to address the mentioned issues but in a limited way especially in respect of real world applications. Besides, some works proposed to overcome performance issues by increasing the considered number of sensors. However, even if adding more sensor to different on-body positions goes hand in hand with an increased performance [4, 40], it does not solve the outlined problems. Besides, most researchers of this domain usually try to rely on a minimal setup by trying to achieve sufficient results.

1.2.3 Activity Recognition with External Sensors

While the physical activity is a valuable information concerning physical exercises, it says nothing about the used objects or the person's location meaning the overall context is unknown. As a consequence, important activities (ADLs) are not recognized which are critical in our scenario but also for most pervasive computing systems. For that reason, researchers started to focus on smart-environments or smart-homes, i.e., flats or apartments that are equipped with external sensors. Those sensors are attached to items, furniture, or walls to capture the mentioned aspects. The general idea is to recognize the activity that triggered the reported sensor events by analyzing and associating those sensor events.

Similar to physical activity recognition, supervised learning was proved to be effective but its applicability to complex ADLs in a real world scenario is questionable. As there are significantly more possibilities to execute an ADL than a physical activity, it would be necessary to acquire a large dataset of ADLs to capture most execution patterns in different situations. Further, activity execution patterns are strongly coupled to a person's characteristics and environment, and the portability of activity datasets is an open issue [41]. For that reason, ideally one extensive ADLs dataset should be acquired from each monitored user. Unfortunately, acquiring ADLs datasets is very expensive in terms of annotation costs [42, 43] and an external observer, e.g., cameras or direct observation, would even violate the user's privacy. To overcome that problem, other works relied on knowledge-based activity models, manually specified through logic languages and ontologies. Those models are matched with acquired sensor data to recognize the activities [44–46]. However, the main shortcoming of that approach relies in the rigidity of specifications. For instance, complex ADLs are often specified through temporal sequences of simpler actions [47]. Nevertheless, it is unfeasible to enumerate all the possible sequences of actions describing a complex ADL.

Several pervasive computing applications already call for *online* activity recognition systems, i.e., systems that can recognize the current ADL in nearly real-time [48]. For instance, a system to detect dangerous behaviors of the user should report the potential danger as it happens, since a delay could put the user's safety at risk. Unfortunately, several proposed ADL recognition systems are limited to offline recognition, and the accuracy of real-time ADL recognition systems is generally lower than those of offline ones [49].

Active learning has been proposed to mitigate these problems, hence, it reduces the need of a comprehensive dataset or improves the performance of an online based system as the technique collects information in real-time to adapt the system at runtime. However, the majority of these techniques need anyway a starting labeled training set. Alternative approaches propose the use of a structured knowledge representation of activities, infrastructure, and events to guide the recognition process in an unsupervised way [2]. In order to be effective, they require a significant effort of knowledge engineers to build a comprehensive ontology, and it remains questionable if such an ontology could actually cover a heterogeneous large set of environments and individuals.

Beside problems related to recognizing ADLs, there are also several open issues that go along with the infrastructure of a smart-environment. This includes the assumption about the consistency of the underlying sensor network but also the number of people in this area. Hence, a smart sensor network has to be adaptive meaning it has to deal with new installed or failed devices [50]. Further, especially without cameras it is not trivial to identify which sensor event was triggered by which person. Indeed, that an apartment is inhabited by several people seems to be the general case but most existing approaches focus on a scenario with just a single resident [51, 52].

1.3 Research Questions

Our aim is to develop robust activity recognition methods based on external and wearable sensors that generate high quality results in a real world setting. In order to achieve this, we focus essentially on the problems that emerge when coming out of the lab. Thus, we are mainly interested in finding feasible solutions for those problems and getting useful insights for implementing corresponding techniques. As before, we outline our research questions in respect of wearable (**RQ1.x**) and external (**RQ2.x**) sensors, overall aiming to create a sound basis for a hybrid solution. Hence, in this work we focus on bringing physical activity recognition but also the recognition of ADLs out of the lab to support diabetes patients regarding physical exercise and activities of interest. However, we are convinced that on a long term a hybrid based solution would help patients even more, i.e., relying on external and wearable sensors simultaneously.

In particular, we want to answer the following questions:

RQ1.1 Is it possible to recognize automatically the on-body position of a wearable device by the device itself?

RQ1.2 How does the information about the wearable device on-body position influence the physical activity recognition performance?

RQ1.3 Which technique can be used to build cross-subjects based activity recognition systems?

RQ1.4 Given a cross-subjects based activity recognition model, how can we adapt the model efficiently to the movement patterns of the user?

RQ2.1 Which method can be used to overcome the requirement of a large expensive labeled dataset of Activities of Daily Living?

RQ2.2 Which type of recognition method is suitable for handling the diversity and complexity of Activities of Daily Living?

RQ2.3 How can external sensor events be exploited to recognize Activities of Daily Living in almost real-time?

RQ2.4 Given a generic model of a smart environment, how can it be adapted to a certain environment and user at run-time?

[RQ1.1] and [RQ1.2] are directly connected and focus on a common problem in respect of activity recognition approaches which use wearable devices. In earlier works, researchers proved under certain conditions the reliability of recognizing physical activities by inertial sensors (mainly accelerometer). Today, many researchers motivate this kind of works by referring to wearable devices which feature such sensors and which facilitate to apply physical activity recognition in everyday life. However, the influence of the on-body device positions is often ignored and so far, nobody investigated the feasibility concerning all relevant on-body positions and physical activities. Moreover, it is even unclear whether the on-body device position can be recognized to handle upcoming position changes of the device. Similarly, [RQ1.3] focuses on overcoming a problem that goes along with a common scenario in particular to support elderly or patients. For one, it is often not feasible that these people collect and label required data and if the system is required, it should be immediately at hand. For that purpose, we focus on identifying a suitable approach that enables to use existing data also for new users. Indeed, several works already concluded that a single-subject based approach performs the best. For that reason, [RQ1.4] is concerned with the aspect on personalizing a recognition system at run-time meaning to adapt the recognition model to the behavior of the user. This would also ensure that the performance of the recognition system remains stable on a long term, as it is able to react to changes caused by age or disease.

A major difference between physical activities and ADLs is the level of complexity, thus it might be feasible to collect and label required data in respect of the former but not for the latter (cf. [53]). Therefore, [RQ2.1] deals with the problem of using the recorded sensor data in an unsupervised way, i.e., exploiting possible correlations between sensor events and ADLs. Additionally, [RQ2.2] addresses the question of identifying a suitable technique that is able to handle such correlations and which is flexible in recognizing varying ADLs. In this context, we mainly focus on a probabilistic-based approach as it has several advantages compared to specification-based or classical machine learning based approaches. Related to [RQ1.3], the questions which arise is how to implement or apply the identified solutions in an online fashion ([RQ2.3]). In the best case, the ADL is recognized as fast as possible to react to the current situation. This requirement presupposes that the recognition system is able to detect transitions between ADLs but also to recognize the ADL while it is performed. For that purpose, we focus on finding a suitable strategy that encapsulates sensor events that describe the same activity. The last research question ([RQ2.4]) goes also hand in hand with [RQ1.4], i.e., we focus on how to personalize and adapt the system that results from [RQ2.1]-[RQ2.3] to upcoming changes.

1.4 Contribution

Along with answering our research questions (**RQ1.x** and **RQ2.x**), we also contribute to the field of pervasive computing and communications. This includes a new dataset, novel methods, and comprehensive empirical investigations in respect of recognizing activities with sensors. In the following, we summaries our main contributions.

Activity Recognition with Wearable Devices (Chapter 4)

- We present a new real world dataset for on-body position detection and position-aware physical activity recognition.
- We show that our on-body position recognition method consistently improves the recognition of physical activities in a real world setting.
- We show that using labeled data of different people of the same gender and with a similar level of fitness and statue is feasible for cross-subjects activity recognition for people that are unable to collect required data.
- We perform comprehensive experiments regarding cross-subjects models in context of offline and online learning with single and multi-acceleration sensor setups including all common physical activities and on-body positions.
- We present a physical activity recognition approach that personalize cross-subjects based recognition models by querying the user with a reasonable number of questions.

Activity Recognition within Smart Environments (Chapter 5)

- We present a novel unsupervised Activity of Daily Living recognition method that overcomes the main drawbacks and limitations of supervised- but also specification-based approaches.
- We explicitly handle and recognize interleaved activities, while many other works are restricted to sequential ones.
- We introduce a novel online segmentation algorithm that combines probabilistic and symbolic reasoning to divide on the fly a continuous stream of sensor events into high quality segments.

- We introduce an approach that it is able to recognize Activities of Daily Living in almost real-time while the recognition quality is close to an approach that runs in offline mode.
- We propose a new active learning approach to Activity of Daily Living recognition that addresses the main problems of current statistical and knowledge-based methods.

1.5 Outline

In the following, we outline the structure of our work and summarize the respective chapters.

Chapter 1: Introduction. The preceding introduction outlines the field Human Activity Recognition and describes in this context the idea of using wearable and external sensor based concepts to support active and healthy ageing. This is accompanied by our research questions and the respective contributions.

Chapter 2: Preliminaries. We introduce preliminaries that are necessary for understanding our approaches, discussions, and conclusions. This includes a short discussion about terminology, relevant parts of existing sensor technologies, and fundamentals in respect of Machine Learning, Description Logic, and Probabilistic Reasoning. In addition, we also clarify the benefits of these techniques in terms of recognizing activities.

Chapter 3: Related Work. We present related work grouped by activity recognition with wearable devices and within smart environments. Our intend is that the reader gets an impression regarding the state-of-the-art, open issues, but also of the research field in general. For that reason, we present a broad view for both parts. As an extension, we discuss further research directions and upcoming issues in the respective chapters.

Chapter 4: Activity Recognition with Wearable Devices. We focus on the introduced open issues in respect of physical activity recognition with wearable devices and present related approaches, solutions, experiments, and discussions. For that purpose, we explain the data gathering process, introduce preprocessing techniques, present an approach that addresses the device on-body localization problem, introduce a position-aware activity recognition approach, investigated the possibility of cross-subjects based recognition models and present a solution to evolve a physical activity recognition model over time. We conclude this chapter with a comprehensive discussion in respect of open issues and further research directions.

Chapter 5: Activity Recognition within Smart Environments. Similar to the preceding chapter, we focus on the introduced open issues in respect of recognizing Activities of Daily Living. We deploy a reliable and feasible recognition system which overcomes common limitations of existing system. For that reason, first we introduce two datasets which we use to evaluate our approach. Then, we explain the required preprocessing steps followed by the explanation of the concept of our approach including online recognition and active learning components. We conclude this chapter with a comprehensive discussion in respect of open issues and further research directions.

Chapter 6: Conclusion and Future Work We conclude our work by recapping and answering our initial research questions but also clarify how everything connects. In respect of future work, we summarizes our preceding discussions and highlight promising research directions.

Chapter 2

Preliminaries

In this chapter, we introduce preliminaries that are necessary for understanding our approaches, discussions, and conclusions. This includes a short discussion about terminology as there is no common agreement within the pervasive computing domain on how to denote common types of activities but also common types of approach (Section 2.1). Subsequently, we introduce relevant parts of existing sensor technologies so the functionality and synergy of sensors which we consider for recognizing activities (Section 2.2). Finally, we introduce fundamentals in respect of Machine Learning (Section 2.3), Description Logic (Section 2.4), and Probabilistic Reasoning (Section 2.5). Here, we focus on the essential idea and the underlying concept but also on related strategies that are applied in this work. In this context, we also clarify the benefits of these techniques in terms of recognizing activities.

2.1 Terminology

Especially terms that should reflect certain types of activities are sometimes used contradictorily or activities of different types are denoted with a single term. Further, different approaches (e.g. focusing on a single user or several users) are often denoted with different terms but having the same meaning. For that reason, in the following we outline terms that are commonly used, their connections, and used synonyms. Subsequently, we specify which terms are used in the remaining of this work where we follow the most common usage.

2.1.1 Activities

Activities can be grouped and denoted based on their complexity level (e.g. physical activities vs. ADLs) and in turn further subdivided by their type (e.g. fitness vs. transportation). Nowadays, existing works focus on recognizing all these types of activities while activities with different complexity levels are frequently intermixed (e.g. [18, 19, 25, 54]). This can be confusing and maybe even misleading. For that reason, we introduce a range of terms which we use in this work but that are also used in related work where their usage can differ across different works. Thus, we would like to clarify how we use these terms where we do not want to introduce a hierarchy of activities but we want to give the reader an idea about how to distinguish between different activities.

Actions, Activities, and ADLs [55, 56]. These are the most common terms for denoting a group of activities where the term *Activity* is usually complemented by *physical*, *simple*, *complex*, *low*, *high*, *micro*, and *macro*. The terms *simple*, *low*, and *micro* and also *complex*, *high*, and *macro* are synonymous usually used for distinguishing between *physical activities* and *ADLs*. In this context, we consider an activity as a physical activity when it is performed without items or interactions. This includes walking and running but also standing and sitting. Hence, *physical activities* are also often denoted as ambulation, posture or locomotion. Comparing the terms *actions* and *physical activities*, the difference is essentially that an action just takes a moment as it is the case for grabbing, hand shaking, or opening a door where physical activities are often cyclic or permanent. In contrast, *ADLs* are characterized in particular by the fact that someone is interacting with items or other people and at the same time pursues a specific goal or acts in respect of a certain context. This includes preparing a meal but also shopping and transportation. Indeed, performing an ADL goes usually along with a physical activity but also with actions. Thus, these groups do not exclude each other but refer to different perspectives of an activity.

Upper and Lower Body [15, 57]. The idea to group activities by upper and lower body results from fact that several activities can be executed by only using certain body parts. Simply put, only the legs are required to walk while the movement of the arms can be arbitrarily. In contrast, if someone is sitting at a table the performed activity is usually characterized by the movement of the head or arms. Thus, this distinction is not bound to a certain activity type and should clarify that it is necessary to capture both body parts to recognize entirely all activities.

Static and Dynamic [58]. These two terms refer to the movement of the human body; hence, if an activity is performed almost without moving as it is the case for standing or sitting it is called a *static activity* where the counterpart (e.g. walking or cycling) is called *dynamic activity*. Indeed, these terms are usually used in respect of postures and movements (ambulatory activities) and are applicable for simplex and complex

Table 2.1: Exemplary overview of commonly used groups for summarizing certain activities. Static and dynamic refer to the necessity of moving the body (or certain body parts). Overall, these groups do not exclude each other but refer to different perspectives and granularity of an activity.

Group	Static	Dynamic		
		Lower body	Upper body	both
Action	inhale, exhale	kick, single step, tread down	grabbing, opening door, pressing button	forward roll, falling, lay down
Physical Activity	standing, sitting, lying	walking, running, climbing stairs	walk on hands, clapping, head-shaking	climbing, star jumps, crawling
ADL	watching TV, reading a book sleeping	go for a walk, cycling, kicking a ball	speaking, drinking coffee, using a PC	shopping, cleaning, driving

activities. Overall, this distinction is often considered as a first step for analyzing physical effort or as a context information of the current situation.

Table 2.1 provides an exemplary overview of these groups aiming to illustrate the relation and overlap of these terms.

2.1.2 Approaches

Independent of the type of activity, there are different approaches in respect of building a model that recognizes activities. Basically, we distinguish between a *single-subject* (also known as user-specific) and a *cross-subjects* approach where the latter can be further subdivided by specifying which subjects are considered. Both have different benefits and differ in terms of required data meaning a single-subject approach only relies on data of the subject for which the model is intended. In contrast, a cross-subjects approach relies on data of several different subjects aiming to build a more generalized recognition model. An obvious advantage is that a single-subject model usually has a higher accuracy in recognizing activities due to its customization; however, that requires to collect data of each subject for which activities should be recognized. This is often not feasible due to the amount of required data, a disability of the user (i.e. data cannot be collected), or the requirement that the model has to be immediately at hand.

As already mentioned, a cross-subjects approach can be considered as a general term describing approaches that rely on data of several subjects to recognize the activities of another subject. In this context, most existing works focus on a leave-one-subject-out approach meaning that the data of n-1 subjects is considered for building the recognition model where the remaining subject is used for evaluating the model. This is repeated n-times so that everyone was used for testing to clarify if it is feasible to generalize the recognition model. Indeed, depending on the considered subjects and used data this might not work due to contradictions. For instance, if an elderly person is running than this might be walking in respect of a child. The same example also holds for people having the same age but differ significantly by weight. For that reason, people might be clustered according to certain criteria where in turn a classification model needs to be built for each cluster. The advantage of a cross-subjects model would be that it can be immediately at hand as a new or unseen person just needs to be assigned to a cluster where usually the performance is worse as it is not customized.

In theory, there are several enhancements that try to overcome this limitation, e.g., by personalization and collaboration. The general idea of personalization is to adapt a recognition model to the behavior of the user. Thus, a cross-subjects approach can be considered as a starting point where over time the user is asked for feedback in respect of the recognition results. Then, the answers can be analyzed to decide how to adapt the model. Indeed, this idea can be also applied in context of a single-subject approach as usually the behavior or movement pattern of a person changes over time. This would also ensure to keep the recognition performance in the end. In contrast, the idea of collaboration is to collect feedback from all people that use the same model. This should help to compare and rate the feedback of the users but also to keep the model generic. Overall, it has the same goal as personalization but in context of a certain group of people.

In the following, we will go into detail by introducing the considered data sources for building such models but also how to develop and adapt such models.

2.2 Sensor Networks

A sensor network usually consists of a large number of sensors which are densely deployed where the network is deployed either inside the phenomenon or very close to it [59]. Thus, a sensor network has the task to capture certain signals or events which represent the phenomenon of interest and which enable to draw inferences about the phenomenon, respectively. This involves challenges like a changing topology as sensors may frequently added or removed, limitations in respect of power consumption and computational capacities, and sensors which prone to failures. As there exists a wide range of sensor types but also areas of application (e.g. health, military, and home) and as a consequence several different settings, in the following we introduce for one thing the sensor types which are considered in this work and for another thing our phenomenon of interest, i.e., the place where the sensor network is deployed and the kind of activity that should be recognized.

2.2.1 Sensor Types

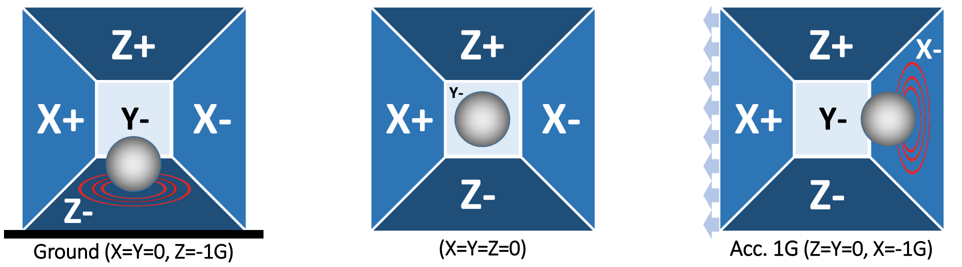
Nowadays, there exists several different groups of sensors including motion, physiological, proximity, and environmental sensors which feature the possibility to monitor completely an individual. In this work, we focus mainly on motion sensors as they are unobtrusive, need low energy, and protect the users' privacy, as they do not record any video, audio, or physiological information. Indeed, motion sensors capture different kinds of motions and movements but also the sensors orientation or orientation changes. In the following, we only introduce sensors which are considered in this work be it for experiments or discussion. More precisely, we introduce the accelerometer, gyroscope, and magnetometer which are embedded nowadays in almost each wearable smart device. Please note that we always refer to a 3-axis implementation of the respective sensors. As we mainly focus on the accelerometer, we introduce this sensor in more detail to clarify how we take advantage of it. Overall, the explanations are independent of smart devices and should help in understanding our argumentation and conclusions.

2.2.1.1 Accelerometer

The accelerometer belongs to the group of inertial sensors and measures the acceleration of a body reflecting the change in velocity for a certain duration of time. From a physical point of view, the laws of motion [60] which were compiled by Isaac Newton describe acceleration (a) as the amount of force (F) that is required to move each unit of mass

(m) ($a = \frac{F}{m}$). Thus, the acceleration is not determined by measuring how the velocity changed over time but by measuring force. Simply put, how much a body (m) presses on something when a force (F) acts on the body (see Figure 2.1). In this context, the acceleration (a) is indicated in $\frac{m}{s^2}$ where s is seconds, the force (F) is indicated in $\frac{kg*me}{s^2}$ where me is meter, and the mass (m) is indicated in kg.

Considering the gravity of Earth, the mean gravitational acceleration is $9.81\frac{m}{s^2}$ that is abbreviated as 1G. An accelerometer at rest relative to the Earth's surface still measures an acceleration of 1G (see Figure 2.1a) where in turn the acceleration is zero when the accelerometer is in free fall (see Figure 2.1b). For that reason, a distinction is made between the gravitational force and the linear acceleration; hence, the latter is the real acceleration of a body where the gravity was eliminated (see Figure 2.1c). The coordi-



(a) Gravitational Force **(b)** Free Fall **(c)** Linear Acceleration

Figure 2.1: Simplified concept of a 3D accelerometer. The figure depicts a ball in a box which presses against a wall of that box depending on how the box is moved. The pressure on the wall and the weight of the ball indicate the acceleration of the box. Thus, when the box stands on the ground (a) then the ball presses on the bottom of the box due to the gravitational force. If on the other hand the box is accelerated in a certain direction (c) then the ball presses on the wall on the opposite side. This is comparable with a human which is pressed back into a car seat when the car speeds up.

nation system of an accelerometer is relative to the body, i.e., independent of the earth coordinate system. Thus, it is not possible to derive information about the direction of movement in respect of cardinal points but to get information about the orientation of the body also known as roll and pitch. This information is derived from the gravitational force and describes the rotation of the body front-to-back (roll) and side-to-side (pitch).

Overall, there are many different types of accelerometers such as mechanical, capacitive, piezoelectric, resistive, and piezo-resistive based implementations. As an example, a capacitive accelerometer consists of two plate capacitors that share a common plate between them. When the accelerometer experiences any acceleration, this common plate is moved which changes the capacity ratio of the capacitors. This change enables to gather the actual acceleration and reflects the concept that is depict in Figure 2.1.

2.2.1.2 Gyroscope

The gyroscope also belongs to the group of inertial sensors and measures angular velocity [61]. This reflects how fast an angular around an axle over time changes and enables to capture the rotation of a body that helps to determine the orientation. A gyroscope can be considered as a symmetrical spinning wheel with a constant angular momentum where the axis of rotation is able to adopt any orientation (movable bearing). Due to the conservation of angular momentum, the wheel has a high persistence meaning when the orientation of the gyroscope changes then the orientation of the wheel remains almost the same. Thus, when a force acts on the gyroscope which affects the orientation and as a consequence tries to tilt the spinning wheel then the axis of rotation tilts perpendicular to the active force to preserve the total angular momentum. Measuring the rotation speed between the spinning wheel and the frame of the gyroscope results in the gyration, i.e., the angular or rotation motion.

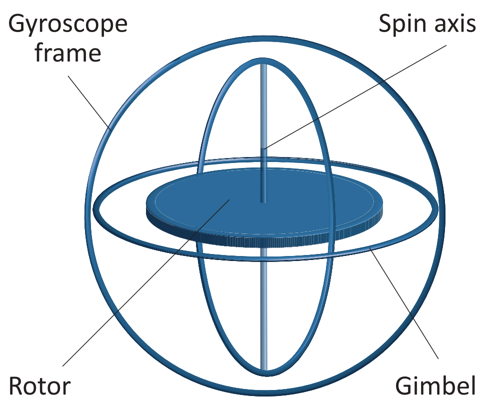


Figure 2.2: Concept of a $3D$ gyroscope. The rotor spins with high and constant speed and as a result caused by the angular momentum the rotors' orientation keeps almost the same when the frame or gimbal is moved.

Figure 2.2 depicts a gyroscope and shows the individual components. For clarification, when the wheel spins with high and constant speed and someone would grep the gyroscope frame and starts walking around then this does not affect the orientation of the wheel, i.e., the orientation of the wheel keeps almost constant. In contrast, the spin axis and the gimbal adapt to the orientation changes which are triggered by walking around. Measuring the shift of these components results in the angular velocity.

Compared to an accelerometer that records the acting force along an axis, a gyroscope is able to capture the rotation of the body. As the accelerometer, the coordination system of the gyroscope is relative to the body, i.e., it has no absolute reference. However, combining both an accelerometer and a gyroscope results in an Inertial Navigation System. Thus, knowing the initial start position and having very high accuracy instruments enables to keep track of the direction of movement. In theory, having only a gyroscope (or accelerometer) and knowing the initial start position would be enough to estimate the movement direction but in practice noise adds up very quickly and the estimation drifts too far away from reality [62].

Nowadays, the term gyroscope is used for variety of rotation rate sensors without a real gyroscope (wheel) but which serve the same purpose as a real gyroscope. Basic types beside the classical rotary gyroscope are *Vibrating Structure* or *Optical* gyroscopes that differ by the implementation of the presented concept and as a consequence have a different accuracy.

2.2.1.3 Magnetometer

In general, a magnetometer measures the strength and the direction of a magnetic field where in turn a magnetic field is a result of moving charges or electrons. Thus, a magnetic field is created by an electric charge or a magnet and in either case the moving particle generate this force field. In case of a (permanent) magnet, the force field spreads between the north and south poles, i.e., each magnet has two poles and thus is a dipole. The force field can be considered as the effective range of a magnet and exerts a force on other magnetic fields but also on elements like iron, nickel, and cobalt. This property is used by an magnetometer to measure the field strength (Ampere/Meter) and the resulting magnetic flux (Tesla) (see Figure 2.3).

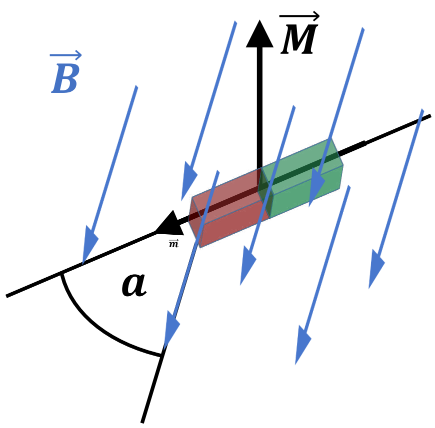


Figure 2.3: Torque on a magnetic dipole. Given the magnetic flux density $\vec{B}$ and the magnetic moment $\vec{m}$ allows to compute the angular α and so the torque. A floating permanent magnet spins until α=0, i.e., points towards north.

The earth has a natural (ambient) magnetic field which is comparable with a magnet and results from the fact that the earth consists in large part of ferric iron. As a compass (which is a simple type of a magnetometer) is nothing else but a magnetic needle, the magnetic forces (Lorentz force) of both fields exerts on each other. As a consequence, the magnetic needle adjusts itself parallel to the field lines of the magnetic flux of the earth (the direction of the ambient magnetic field), i.e., the magnetic field of the earth can be considered as a global coordination system which enables to determine the orientation of a body in respect of the cardinal points (absolute orientation). Overall, a magnetometer takes advantage of these properties to measure the mentioned characteristics of the magnetic field. Indeed, there are several different kinds of magnetism, e.g., ferromagnetism, electromagnet, and diamagnetism; however, we only describe the relevant aspects in respect of this work.

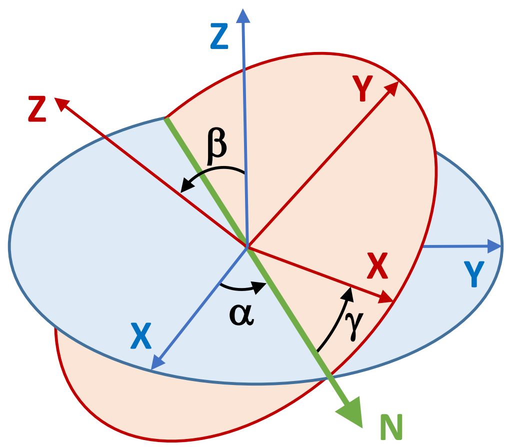


Figure 2.4: Euler angles. These angles were introduced by Leonhard Euler to describe the orientation of a body in respect of a fixed coordinate system. Combining an accelerometer, gyroscope, and magnetometer, these angles can be calculated with high accuracy.

The magnetometer is also often considered as an inertial sensor and so part of an inertial measurement unit. Nevertheless, strictly speaking a magnetometer is not an inertial sensor. Combined with an accelerometer and a gyroscope, it allows to keep track of the orientation of a body for all three dimensions, i.e., it gathers changes in pitch, roll, and yaw (also known as azimuth) (see Figure 2.4). Theoretically, already an accelerometer and a magnetometer are enough to gather these dimensions but adding a gyroscope increases the precision. For example, the accuracy of a magnetometer is poor while moving fast but the accuracy does not get worse over time. In contrast, a gyroscope reacts quickly and accurate to changes but the accuracy drops significantly over time. Further, both the accelerometer and the gyroscope require an initial start orientation as both only react to changes. Hence, these sensors excel each other at different things and combining them allows a quick and accurate position and orientation determination.

2.2.2 Body Sensor Networks

A *Body Sensor Network* (BSN) is a combination of (different) sensors which are directly attached to the body of a human and which operate independent of the user's location. Thus, all components that are necessary for the operation are carried by the user. The concept of a BSN is not limited to certain sensor types but it aims to gather environmental information (e.g. by a microphone), physiological information (e.g. by an ECG), and physical information (e.g. by an accelerometer). In context of *activity recognition with wearable devices*, we rely on such a system to gather physical information about the user, i.e., the movements of the individual body parts to recognize simple activities. Indeed, at present is seems not feasible to carry sensors, e.g., at each body part without restriction of the range of motion. However, progressive miniaturization and upcoming approaches like smart clothes [63] where sensors are integrated in cloth seems to make this feasible. Besides, we are primarily interested in answering the question if this is even meaningful. Overall, in this work we focus mainly on a body sensor network consisting of accelerometers represented by linked wearable devices.

2.2.3 Smart Environments

Nowadays, the term *Smart Environment* or *Smart-Home* are commonly known due to commercial produces like smart voice services (e.g. Amazon Alexa), smart lamps (e.g. Philips Hue), robot vacuum cleaner (e.g. Neato), smart locks (e.g. Nuki), and many more. For that reason, we believe it is necessary to clarify what we mean by *activity recognition within smart environments*. In this work, we do not consider any of these commercial produces but aim to equip everyday objects and furniture in a common home with unobtrusive sensors that are able capture the performed activity. For instance, accelerometers can be attached to objects of interest or doors to register when they are used. Hence, one focus is to keep the sensor network passive, i.e., the resident does not have to interact consciously with any device. That is especially important as such a system should have a certain reliability which is independent of the mood of the resident. Besides, a local sensor network is independent of external (commercial) services and easily adaptable to other homes. Overall, when we use the terms smart environment or smart-home, we refer to this description.

2.3 Machine Learning

The field of *Machine Learning* (ML) belongs to the *Artificial Intelligence* area and describes groups of approaches that in general try to *learn* behavior or patterns from data or which aim to gain new or hidden knowledge from data. The most and well-known groups are classification, clustering, regression, and association rules which are intended for solving different problems or addressing different use cases. For example, naturally classification techniques are used to decide to which set of categories a new observation belongs while clustering techniques aim to group observations in such a way that similar observations are in the same group.

Overall, this section is only intended to introduce the preliminaries in respect of ML techniques that are applied in this work. Hence, we only focus on the relevant aspects with respect to this work. In this context, we define the terms *supervised*, *semi-supervised*, and *unsupervised* learning concerning classification-based approaches. Subsequently, we introduce relevant classification algorithms, i.e., their way of functioning as well as advantages and disadvantages.

2.3.1 Classification

Classification can be considered as the task to classify a single data record with a predefined class or label. This data record can be considered as a description of an instance

Table 2.2: An excerpt of an ADL training dataset. It consists of a set of features and a set of class labels where each row can be considered as a training sample for building a classification model.

Features					Class
Location	**Daytime**	**Interaction**	**Weather**	**Posture**	**Activity**
Kitchen	Midday	Knife	Sunny	Sitting	Eating
Kitchen	Midday	Knife	Rainy	Standing	Preparing Meal
Living Room	Afternoon	Bowl	Cloudy	Sitting	Watching TV
Kitchen	Morning	Spoon	Rainy	Sitting	Eating
Living Room	Afternoon	Cloth	Rainy	Walking	Cleaning
Kitchen	Morning	Water	Sunny	Standing	Cleaning

or state, i.e., a set of features which characterize the target class. In this context, Table 2.2 depicts a simple example where each column except the last can be considered as a feature that should be predictive for the class in the last column. The goal is to find a classification function which is reliable in recognizing the target class based on the available features. The challenge is to find a function which does not overfit meaning it works only in respect of the available example or training data but has a significant drop in reliability classifying new or unseen data records.

Figure 2.5 shows a classical but also simplified process for building a classification model. The dataset (step 1) is the basis for the succeeding steps and needs to be split into training (step 4), testing (step 5), and validating (step 6) datasets. The training dataset is a set of examples (cf. Table 2.2) that is used for learning a classification model, i.e., to find a function that derives from the given features the corresponding target class. Subsequently, the testing dataset is used to measure the reliability of the model. Usually, the training and testing datasets have to be disjointed but should have the same distribution in respect of the samples of the respective classes. If a model fits to the training and testing data then probably minimal overfitting has taken place. In contrast, when the model fits the training data but not the testing data then this usually points to overfitting. Finally, the validation dataset is used to tune the hyperparameters, i.e., the architecture of a classifier. Considering an artificial neural network, a hyperparameter is the number of hidden units. Some simple algorithms including ordinary least squares regression does not have any hyperparameters. Besides, the validation dataset should also have the same distribution of samples as the training and testing dataset.

Formally, we define a classification problem as follows: Given a fixed set of classes $C = \{c_1, c_2, \ldots, c_n\}$ where we denote by class an abstract concept and by instance the

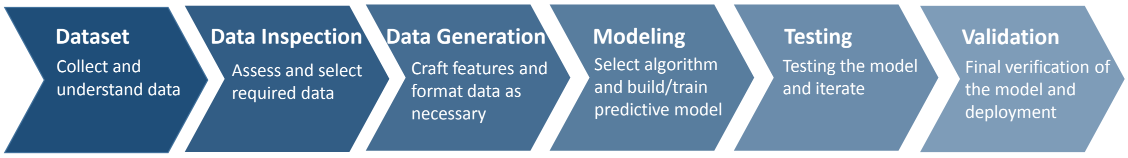


Figure 2.5: Generic process for learning and validating a classification model.

actual occurrence of that concept. A description $d \in X$ of an instance where X is the set of all possible descriptions for all instances. A training dataset S and a testing dataset T of labeled instances with each instance $\langle d, c \rangle \in X \times C$ where $S \cap T = \emptyset$. Then the goal is to find a function $f : X \to C$ by using S which is especially reliable in assigning $\forall \langle d, c \rangle \in T, f(d) = c$.

2.3.2 Types of Learning

In context of ML, there exist several different learning strategies or algorithms target different kind of problems and aspects. Figure 2.6 provides a rough categorical overview.

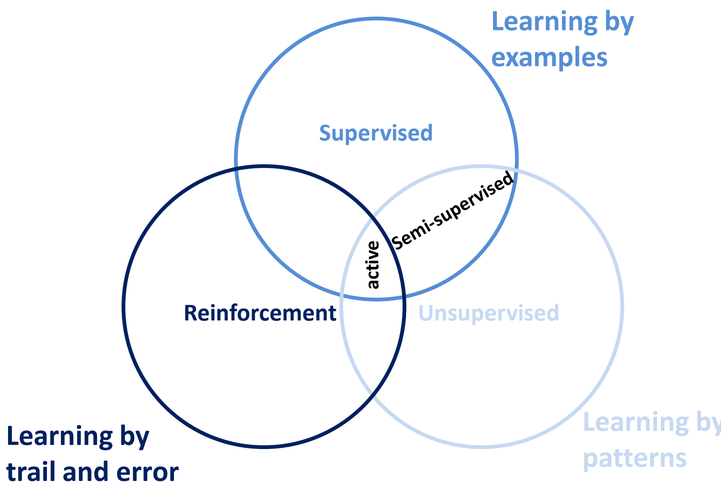


Figure 2.6: The required type of learning usually results from the learning problem, i.e., whether the available data is labeled (supervised), unlabeled (unsupervised), or there is no data available at all (reinforcement).

Simply put, supervised and unsupervised learning describe whether the available training data is labeled or not. For instance, considering Table 2.2, the last column covers the corresponding labels for each row; hence, considering this column makes the difference between supervised and unsupervised learning. Indeed, this means that a classical classification approach goes always along with a supervised learning strategy. If, however, one wants to recognize the correct class without labeled data, i.e., in an unsupervised way then association rule learning might be applied. The idea is to create or construct rules that reflect correlations or relations between events or signals that occur close in time to identify the corresponding class or category. For that, hidden relations or structure must be identified or domain experts have to model the scenario as it is, i.e., independent of the data. As this might suggest, association rule learning can be used in an unsupervised but also supervised way. Usually, labeling data involves a lot of effort and in addition, it is in some real world scenarios not feasible. For that reason, an unsupervised based approach can be a solution to overcome this problem. However, comparing supervised and unsupervised based approaches often shows a gap in performance, so the former reaches a better accuracy.

As a tradeoff, a semi-supervised learning strategy uses both, labeled and unlabeled data. Typically, this strategy requires a small amount of labeled data and a large amount of unlabeled data aiming to have the advantages of both, supervised and unsupervised learning. The underlying idea is to build, e.g., a classification model based on the labeled data which processes the unlabeled data to identify uncertainties in respect of the classification result. These uncertainties help to identify descriptions of instances that have a high information gain, so which have the greatest benefits in improving the classification model. In this context, a common strategy is active learning which involves the user. Thus, after identifying descriptions where the classification result had a high uncertainty, the user is queried to inquire the correct label of that description. The user's answer is usually associated with the description without further evaluation for updating the classification model.

Especially the latter aspect is part of a reinforcement learning strategy. In other words, this strategy accepts correct and incorrect labeled descriptions and tries to maximize the obtained reward (information gain). The idea is to make a tradeoff between exploration of unlabeled data and exploitation the already gained knowledge to learn and reflect the real behavior [64]. Indeed, there exists several different algorithm which implement this concept in different ways, however they are not part of this work.

2.3.3 Offline, Incremental and Online Learning

It depends on the situation or scenario when (training) data become available. For that reason, there are different approaches how to process or handle training data for building or evolving, e.g., a classification model, namely offline, incremental and online learning.

Offline learning, also often called batch learning, consumes and analyzes all available training data to find a reliable function. The advantage is that the data is stored and can be accessed repeatedly where usually most patterns are reflected by the data. However, after the training phase has been completed, the model or function is static meaning for reacting to changes in the patterns, the classification model has to be retrained from scratch.

In contrast, incremental learning is a dynamic technique that continuously update a classification model where initially none or only a small amount of training data is available. The model is updated as data arrive, i.e., it processes a single training sample at a time and caches preceding samples for analyses. Thus, the model does not need to be retrained when patterns change. This is especially useful for adaptive systems or infinite data streams. However, a common drawback of this approach is a lower quality of learning results.

Online learning is similar to incremental learning but it discards a new training sample immediately after it was processed. This is particularly useful when recent data is more important than older but also helpful in respect of the amount of information and disk space when having an infinite data stream. In general, it depends on the implementation of the classifier how to handle this internally. Usually, it temporally caches the information or keeps statistics until it reaches a critical mass to make a decision how to change or extend the model.

In the following, we will go into detail and introduce a range of offline but also an online classifiers that are used in this work. This should give an idea how the classification functions or models are build but also help in understanding results and discussions.

2.3.4 Classification Techniques

In general, a distinction is made between binary and multiclass classification, i.e., if a sample (instance) has to be classified (labeled) with one out of two (binary) or one out of several (multi) class labels. Hence, certain classification techniques can only handle a binary classification problem where usually also these kind of algorithms can be applied in respect of a multiclass classification problem by various strategies. In this context, multiclass classification should not be confused with a multi-label classification problem as the latter refers to the problem of classifying the same sample with several class labels. In the following, we only focus on a multiclass classification problem.

Beside binary and multiclass classification, classification techniques commonly also differ in how they handle *outliers* and avoid *overfitting*. The term *outliers* (or anomaly) refers to samples that are part of the training data and do not conform to an expected pattern or to the remaining observations (samples). Thus, typically outliers are misleading and so have a negative influence on the result, i.e., a less accurate model and with that poorer classification performance. In contrast, *overfitting* describes the problem of creating a classification model which almost perfectly reflects the structure of the training data but usually does not generalize, i.e., it is able to correctly classify the training data while new or unseen samples are often wrongly classified.

There are various methods for evaluating the performance of a classification model. A common approach is n-fold *cross validation* which splits the available labeled data into n-folds where $n-1$ folds are used for training and the remaining one is used for testing. This process is repeated n-times so that in the end each fold was used for testing. Analyzing the results and computing, e.g., the variance across all runs gives some indication about the reliability. Commonly, cross validation is combined with *stratified sampling*, as the number of samples per class might be unbalanced. Thus, stratified sampling ensures that each fold has the same class proportion as the original dataset which in turn makes the results of each run more representative and comparable. In case that the samples per class are balanced so the dataset covers per class round about the same number of samples then random sampling is also a common strategy. Indeed, there are further strategies such as oversampling and undersampling which want to balance the data by generating more samples of minority classes and by removing samples of majority classes, respectively.

In the following, we introduce a range of classification techniques that we consider in our experiments where we will discuss later on recent classification techniques such as

LightGBM [65] or Deep Learning [66] as they were not considered in our experiments due to various reasons.

2.3.4.1 Naive Bayes

The *Naive Bayes* classifier [67] bases on the Bayes theorem, is one of oldest approaches, and is often considered as a baseline method. The Bayes theorem [68] describes how to compute a conditional probability $P(B|A)$, i.e., how likely it is that a certain event B happens, given event A has already happened. In other words, how likely it is that a certain activity C_i took place (event B) given a set of feature values X (event A). In this context, the Bayes theorem is defined as follows:

$$P(C_i|X) = \frac{P(X|C_i) * P(C_i)}{P(X)} \tag{2.1}$$

where $P(X)$ is constant for all classes, i.e., only $P(X|C_i) * P(C_i)$ needs to be maximized. Thus, if the classifier has to classify an unseen record (set of feature values) X then it computes for each considered activity $P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$ to determine (with the Bayes theorem) the activity C_i with the highest posterior probability. This means the classifier computes for each individual feature value $x_k \in X$ the probability that it happens given that C_i happened based on the frequency of that combination in the training data. Hence, the probabilities for each feature value are computed independently of the other feature values, i.e., the *Naive Bayes* classifier assumes that features are (strongly) independent.

For instance, considering our training dataset example (see Table 2.2), the unlabeled record X={Kitchen, Afternoon, Spoon, Rainy, Sitting}, and the activity *Eating* then *Naive Bayes* would compute for each value of the unlabeled record how likely it is that they occur while *Eating* is performed, i.e., $P(Kitchen|Eating) = 1.0$, $P(Afternoon|Eating) = 0.0$, $P(\mathit{Spoon}|\mathit{Eating}) = 0.5$, $P(\mathit{Rainy}|\mathit{Eating}) = 0.5$, $P(\mathit{Sitting}|\mathit{Eating}) = 0.0$ resulting in $P(X|Eating) = 0.0$ as the training data does not cover the event that it is *Afternoon* while *Eating*. This shows clearly why this is a naive approach. In contrast, the Naive Bayes classifier stands out due to very short calculation times.

2.3.4.2 k-Nearest Neighbors

The k-Nearest Neighbors classifier [69] (k-NN) also belongs to the simplest approaches and classifies an unseen record by identifying the k nearest (most similar) training samples (neighbors) by comparing the individual features. The classification result is computed based on these k training samples (also called instance-based learning) by applying a majority vote on their labels where the samples can be weighted based on their distance. Hence, there is no real training phase (lazy learning) as the distances cannot be precomputed meaning the training data is not generalized. In this context, the dimension and data type of the features can differ so different metrics are required to compute the distance between the training samples and an unseen record. Thus, k-NN is based on feature similarity where usually numerical values are compared by the Minkowski distance and strings by the Hamming distance or more complex string metrics like the Levenshtein distance or the Jaccard similarity. Formally, k-NN can be defined as follows:

$$S_k(S, d) = \underset{k}{\operatorname{argsort}}\{\forall s \in S | dist(d, s)\} \tag{2.2}$$

where S is the training set, d the unlabeled record, *dist* is a proper distance function, and *argsort* returns the reference of the k nearest samples in S. This allows to compute the class of d as follows:

$$c(k, S, d) = \underset{c_j \in C}{\operatorname{argmax}} N(S_k(S, d), c_j), \tag{2.3}$$

where $N(S_k(S, d), c_j)$ computes the number of members that were returned by $S_k(S, d)$ and belong to class c_j. Referring this definition, Figure 2.7 shows a simple example to clarify meaning of these parameters. Considering our training dataset example (see Table 2.2), we would need to define or use distance functions that describe the similarity between daytimes (e.g. *Morning* vs. *Afternoon*) but also objects (e.g. *Knife* vs. *Bowl*).

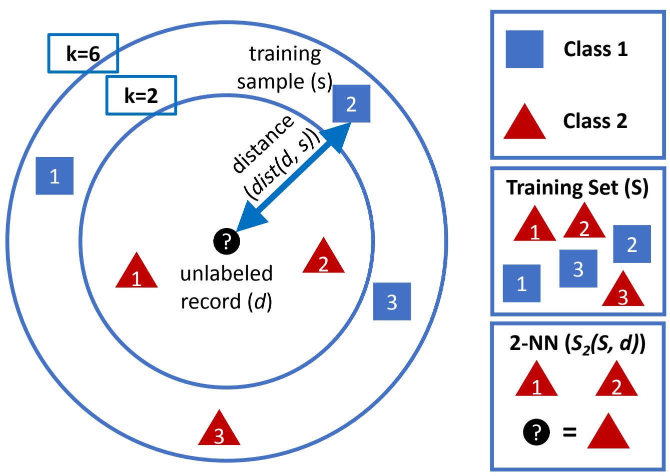
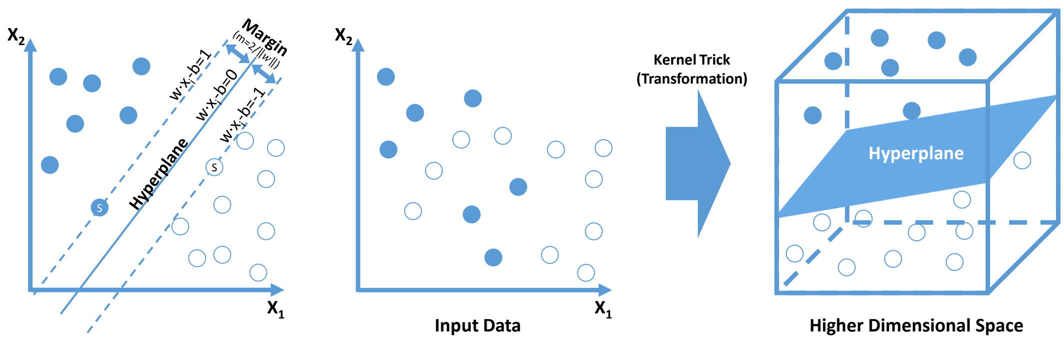


Figure 2.7: Simple classification example using k-NN. The unlabeled sample is compared with all available training samples where the k most similar (smallest distance) ones are used for determining the class label simply by performing a majority vote.

As this shows, k-NN is insensitive to outliers as outliers usually have a high distance that in turn means that those are not considered as *Nearest Neighbors*. However, this approach is (very) sensitive to irrelevant features as all features are taken into account to compute the distance between to samples.

2.3.4.3 Support Vector Machine

A Support Vector Machine [70] (SVM) computes a hyperplane that separates the training samples by class. Therefore, the challenge is to find a reliable hyperplane which separates the data but is not overfitted. In this context, each training sample is represented as a vector in an n-dimensional space where the SVM tries to determine a hyperplane which maximizes the margin to the different classes (see Figure 2.8a). As a plane is flat and not all data is separable by a plane, the SVM usually transforms the input data to a higher dimensional space to make them linearly separable (see Figure 2.8b). For that purpose, the SVM uses a technique called kernel trick where the transformation function is called kernel. After the hyperplane was computed, the data is transformed back to the initial dimension and consequently, the linear hyperplane is converted into a non-linear hyperplane.



(a) Separation hyperplane **(b)** Kernel Trick

Figure 2.8: Simple classification example using a SVM. A SVM aims to identify an optimal separation hyperplane (left) which separates the training example by class label. In case that the data are inseparable by a plane, the classifier transforms the input data into a higher dimensional space using a kernel trick (right) to make them separable.

Finding the optimal separation hyperplane is an optimization problem where a SVM uses quadratic programming to satisfy the following constraint (cf. Figure 2.8a) for any sample (x_i, y_i) in the training dataset:

$$y_i(w * x_i - b) \geq 1 \tag{2.4}$$

where w is the width of the margin. Given that we found an optimal separation hyperplane by checking for each training sample the condition stated above then the vectors of each class (training samples) with the shortest distance (margin) to the hyperplane become *support vectors* (see Figure 2.8a). Indeed, the concept of *support vectors* enables to formulate the problem as an optimization problem of finding the *maximum-margin hyperplane*. Thus, majority of the training data can be ignored which increases the computational speed and makes it robust to outliers. Besides, having a hyperplane with a high margin is an indicator for robustness.

A SVM also has a range of tuning parameters that influence the resulting hyperplane. Beside the already mentioned kernel function (e.g. linear, polynomial, or exponential), one can also specify the range of influence of a single sample (often called *gamma* parameter) which in turn affects the choice of the support vectors. Further, a regularization parameter defines how much to avoid misclassifying the training samples as it is not always meaningful to force a perfect separation.

Having a multiclass classification problem, there are several solutions how to apply the concept of a SVM like *one-against-one* or *one-against-the-rest* [71]. Now if an unlabeled record needs to be classified, the position of this record (vector) in relation to the hyperplane results in the target class.

2.3.4.4 Artificial Neural Network

Artificial Neural Network [72] (ANN) is a general term that covers multiple types of neural network classifiers, differing by their learning strategy, level of complexity, and intended use case. As the name suggests, the concept of ANNs was inspired by biological neural networks that constitute brains in respect of information processing and modeling. Simply put, an ANN consists of several nodes grouped by three types of layer, namely input, hidden, and output. Every ANN has at least one input layer that covers several neurons (nodes) usually representing features derived from the training data. The same is true for the output layer, i.e., there is at least one output layer and the neurons represent the class labels. However, the hidden layer types vary (e.g. long short-term memory, fully connected, or convolutional) and are chosen depending on the data type (e.g., time series or images). Further, the neurons in each hidden layer are determined during the training phase. These neurons represent functions which should map or transform the input to the output where the neurons of the input and output layer are connected with these neurons by weighted links (see Figure 2.9). A weight determines the influence of a function and it is calculated during the training phase.

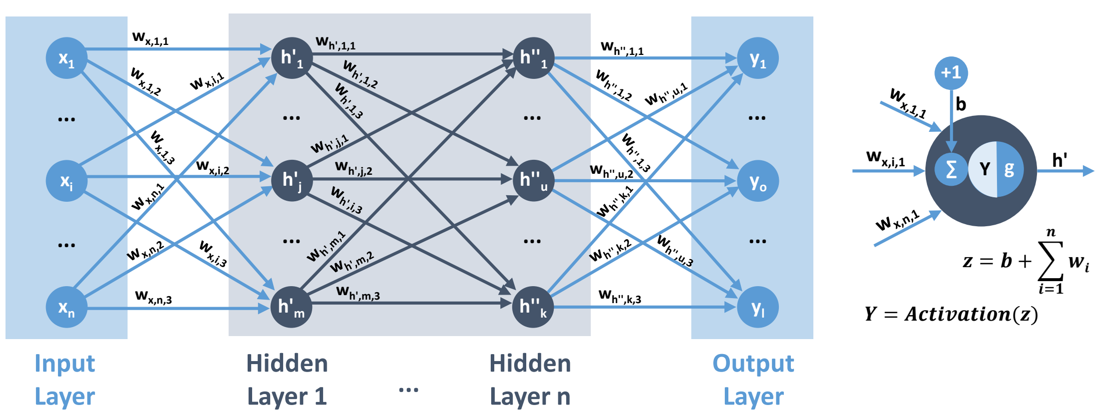


Figure 2.9: General concept of an Artificial Neural Network. The input layer represents features that were derived from the training data while the output layer reflects the considered class labels. The hidden layers are constructed during the training phase and they should map the input to the output.

In this context, the type of function represented by neurons needs to be predefined. This function is usually called activation or transfer function it and defines how the input of a neuron is transformed. Usually, the neurons of the same layer also have the same function where the function is chosen dependent on the properties of the (classification) problem (e.g. Sigmoid, Tanh, and ReLU are commonly used). The general concept of a neuron can be defined as follows:

$$Y = \text{Activation}(\sum(\text{weight} * \text{input}) + \text{bias}) \tag{2.5}$$

Hence, the function has to decide if a neuron should be activated or not, i.e., whether the input might be relevant or should be discarded. Thus, an ANN has to identify suitable functions (e.g. by finding correlations or patterns) and weights for mapping the input to the output data where usually more training data result in a more accurate classification model. This is comparable with a human which has to learn a process and that may identify after a while patterns or rules resulting in getting more efficient and making fewer mistakes.

Basically one can distinguish between two strategies, *feed-forward* and *feed-back*. The former can be considered as the simplest type of ANN as it is unidirectional, i.e., the connected neurons do not form a cycle or loop and the information is only forwarded. In contrast, *feed-back* based ANNs have cycles to backpropagate information, e.g., usually the error is computed in the output layer which can be distributed back to optimize the network. This kind of ANNs are also known as recurrent neural networks.

As this brief introduction suggests, ANNs are complex and powerful but also have disadvantages that have to be considered. This includes that hidden layers are a kind of black boxes where it is difficult to understand what happens; hence, it is difficult to influence the construction of the connections and neurons. Usually, the hyperparameters needs to be modified and the output needs to be measured or analyzed to estimate the quality of the hidden layers. Indeed, there are many approaches how to interpret and explain deep learning models but commonly they go along with a lot of (engineering) effort [73]. Finally, the amount of training data has a significant influence on the quality of the classification model, which makes ANNs classifier less suitable for scenarios with less training data. For instance, nowadays especially deep neural networks (i.e. an ANN with many hidden layers) are used for solving many classification problems, as they are feasible and often have a high accuracy but especially those deep neural networks need many training data.

2.3.4.5 Decision Tree

A Decision Tree [74] classifier is a simple but effective classification technique that is implemented as a graph consisting of nodes and directed edges having a tree structure. Each node represent a certain feature derived from the training data where the outgoing edges represent corresponding value ranges. Hence, an unlabeled sample is passed from the root node to a leaf node where at each node the sample is compared with the feature of the node. Depending on the corresponding feature value of the sample, it is forwarded to the corresponding child node until reaching a leaf node (see Figure 2.10). Finally, each leaf node represents a certain class label that in turn is the classification result of the sample. Indeed, this structure and representation makes the classification model completely transparent and easily allows to investigate the decision process.

Training Data

Room	Posture	InUse	Eating?
Kitchen	Sitting	Fork	Yes
Kitchen	Standing	Fork	No
Kitchen	Standing	Water	Yes
Kitchen	Sitting	Water	No
Dining Room	Sitting	Fork	Yes
Dining Room	Standing	Water	No

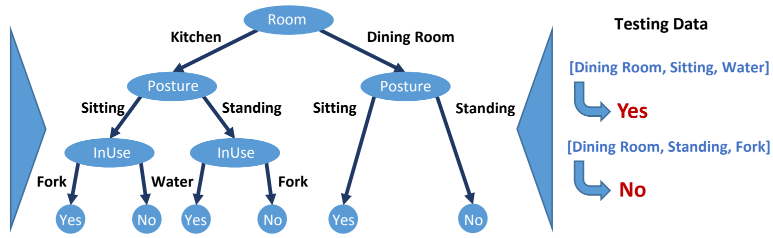


Figure 2.10: A decision tree is built by an iterative process of splitting the training data in partitions at each node until the node is pure or the tree reached a certain depth. An unlabeled sample is passed from the root node to a leaf node based on the feature values where each leaf node represents a certain class label.

Having a training dataset, the construction of the model starts by identifying the feature that best splits the training dataset into two subsets. The selected feature represents the root node and the resulting subsets are passed to the new child nodes. In this context, best split means that each successor node (child) is as pure as possible, in other words a new node should mostly contain samples of a single class. Overall, a node can be considered as a test for the value of a certain feature where depending on a threshold a sample is forwarded to the left or right child of the root node. Thus, the result of this process is a root node representing a test for a certain feature and having two child nodes representing the forwarded subsets. Then, the described process is repeated for each node until it is pure (only covers samples of the same class) or if the tree reached a certain depth. This procedure is also known as the (basic) Divide-And-Conquer algorithm.

Commonly, *Gini Index* and *Information Gain* are used to identify the splitting features; hence, these split functions compute how important a given feature is for predicting a class. In the field, both methods are used and it makes rarely a difference on the classification performance which is chosen [75]. Considering the *Information Gain*:

$$\text{InformationGain(S,F)} = \text{Entropy(S)} - \sum_{v \in \text{Values(F)}} \frac{|S_v|}{|S|} * \text{Entropy}(S_v) \tag{2.6}$$

$$\text{Entropy(S)} = -\sum_{i=1}^{|C|} * \text{P(i)} * \log_2 * \text{P(i)} \tag{2.7}$$

where S_v is a subset of our training dataset (or a forwarded subset) S, F is a feature having a value v, and C is the set of considered class labels. Then at each node for each available feature, the information gain is computed where the feature with the highest information gain is chosen as the splitting feature. Indeed, considering numerical values it is necessary to partition values (e.g. split by a threshold) as the individual values might be less significant in respect of the target class. As a consequence, this also means that the information gain of a feature depends on the chosen partitioning which in turn entails that for identifying a meaningful partitioning, the information gain needs to be recomputed several times for the same feature while adapting the partitioning.

Overall, a decision tree classifier is intuitive, effective, and transparent which can be mainly contributed to the tree structure that is suitable to model dependencies of features but also the straightforward visualization. On the other hand, a decision tree tends to generalize often poorly and to overfit as it becomes too deep. Different techniques of pruning which reduce the depth try to counteract these symptoms but this has to be done carefully.

2.3.4.6 Random Forest

Decision trees have already successfully applied in various domains; however, as mentioned, classical decision trees are sensitive to overfitting when the generated trees become very deep. In order to overcome the overfitting problem, ensemble methods have been proposed that balance the results of multiple decision trees that have been trained on different parts of the training data. Random forest classifiers are one of these ensemble methods that have been proposed by Breimann [76]. As especially Random Forests are usually used in context of activity recognition achieving very well results [14, 18], we are using and introducing this classifier in context of offline and online learning strategies. The latter is considered for clarifying the benefit of online learning for activity recognition in respect of our research questions; hence, as the classical Random Forest was already successfully applied we assume that its online version is suitable for answering these questions.

Offline Learning As a Random Forest is an ensemble of randomized decision trees, the construction is similar to an individual decision tree where usually bagging is applied for reducing the variance. For instance, let $D = \{(x_1, c_m), ..., (x_n, c_o)\}$ be a training dataset where $d \in D$ is a sample consisting of a feature vector x_i and a corresponding class label $c \in C$. In a first step, a number of samples $S_1, \ldots, S_m$ are drawn from D using sampling with replacement. More precisely, for each decision tree t_i, the training set is sampled with replacement, so the set keeps the same size but some instances that occur in the original training set may not appear where others could appear more than once. For each sample S_i, a decision tree classifier f_i is trained using a variation of the introduced decision tree learning algorithm that uses feature bagging. This means that for each branching decision in the decision tree construction only a randomly selected subset of feature vectors is taken into account. This is necessary to ensure that the different generated decision trees are uncorrelated [77]. In this context, the decision tree still considers the information gain or Gini index of each feature to determine the importance during the construction.

The resulting set of uncorrelated decision trees can now be used to determine the outcome for an unlabeled sample x'_i based on the principle of bagging. In particular, the

Algo 1 OnlineBagging(R, L_o, d) [78]

for each base model $t_i \in R$, $i \in \{1, 2, ..., T\}$ **do**
  Set k according to $Poission(1)$.
  do k times
    $t_i = L_0(t_i, d)$.
end for

Algo 2 OfflineBagging(T, L_b, D) [79]

for each $i \in \{1, 2, ..., T\}$ **do**
  D_i = Sample_With_Replacement(D,$|D|$)
  $t_i = L_b(D_i)$
end for
Return $\{t_1, t_2, ..., t_T\}$

result is determined by averaging over the predicted results of all individual decision trees as follows:

$$p_R(c|x'_i) = \frac{1}{T}\sum_{k=1}^{T} p_{t_k}(c|x'_i) \tag{2.8}$$

where the resulting class is $C(x'_i) = \arg\max_{c \in C} p_R(c|x'_i)$. For the case of a classification problem, the combined classifier essentially performs a majority vote over the outcomes of the individual decision trees. It has been shown that bagging prevents the overfitting problem as the combination of multiple classifiers has a significantly lower variance than an individual classifier.

Online Learning Considering the Random Forest classifier in online mode, the main differences are the implementation of bagging, i.e. the generation of subsamples used for constructing the individual trees, and the growing of the individual random decisions trees.

It has been proven that bagging improves the predictive power of Random Forests by generating replicated bootstrap samples of the training set D [79]. This requires that the whole training set has to be available at once. Oza [78] introduced an online version of bagging (see Algorithm 1) where the number of occurrences of a sample for training an individual tree is drawn from a Poisson distribution with a constant parameter. This means that the subsample for a tree can be determined on the fly as a new sample becomes available. Oza provides both theoretical and experimental evidence that the results of online bagging converges towards the results of offline/batch bagging (see Algorithm 2).

The growing of an online decision tree based on the concept of an extremely randomized tree. As in the beginning, the complete dataset is not available, split decision are postponed until enough information is available. This is guided by two parameters: the minimal number of samples that have to be seen before deciding and the minimal quality measurement that has to be achieved by the split. In order to be able to construct further the decision tree, statistics about class membership of new samples are propagated through the tree. It provides the basis for computing the quality measurement of possible splits. As these statistics can easily be updated on the fly, the trees are refined as new samples arrive. In order to compensate for changes in the distribution of arriving information, the results can be adapted by deleting trees whose performance degrade with new information.

Saffari et al. combined the introduced concepts, i.e., online bagging, online decision trees, and random feature selection and developed the first publicly available version of an Online Random Forest [80]. They presented experiments, which show that the Random Forest in online mode converged to the results that were achieved in offline mode. Besides, this classifier is implemented in C++. As we want perform activity recognition on wearable devices, i.e. on an Android platform, we reimplemented this classifier in Java. We repeated the experiments performed by Saffari et al. [80] and achieved the same results. Further, we enhanced the original implementation by implementing threading, incremental learning, and information gain as a quality measurement to split nodes. Our implementation is also publicly available[1].

2.4 Description Logics and Formal Ontologies

In computer science, description logics (DLs) [81] have emerged as the state-of-the-art formalism to represent *ontologies*. The formalism of choice is typically OWL 2 [82] which is a general-purpose modeling language for (certain parts of) human knowledge. It enables to formally define a vocabulary in respect of concepts of a domain of interest (e.g. classes), their properties (e.g. object properties but also data types), and the relationships among concepts (e.g. hierarchies). The resulting ontology can be considered as a knowledge base (or graph) consisting of *ABox* (assertional box) and *TBox* (terminology box) statements. TBox statements describe the *conceptualization of a domain of interest* (i.e. the vocabulary of an application domain) and ABox statements can be considered as assertions about named individuals (i.e. the actual use of the vocabulary). Considering the following example, the first line defines a woman as a female person (TBox, logical equivalence) while the second line states that the individual Mary is a female person (ABox). Further, these two statements allow to derive that Mary is an instance of the concept woman. Several operators can be used to declare such (complex) definitions based on simpler ones, including operators for conjunction, disjunction, negation, and universal and existential quantification.

$$\textsc{Woman} \equiv \textsc{Person} \sqcap \textsc{Female}\ (\textbf{TBox})$$
$$\textsc{Female} \sqcap \textsc{Person}(\textsc{Mary})\ (\textbf{ABox})$$

Defining these more formal, a DLs knowledge base is composed by a pair $\langle \mathcal{T}, \mathcal{A} \rangle$. The TBox $\mathcal{T}$ constitutes the terminological part of the knowledge base. The TBox is composed of a set of axioms $C \sqsubseteq D$ or $P \sqsubseteq R$ (*inclusions*) and $C \equiv D$ or $P \equiv R$ (*equality*), where C and D are classes, and P and R are object properties. An axiom $C \sqsubseteq D$ is satisfied by an interpretation $\mathcal{I}$ when $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. An interpretation $\mathcal{I}$ satisfies a TBox $\mathcal{T}$ when $\mathcal{I}$ satisfies all the axioms of $\mathcal{T}$.

[1] `https://sensor.informatik.uni-mannheim.de/#onlineforest`

The ABox $\mathcal{A}$ is composed of a set of axioms of the form $x : C$ and $\langle x, y \rangle : R$, where x and y are individuals, C is a class, and R is an object property. For instance, "MARY : ELDERLYPERSON" denotes that Mary is an elderly person and "$\langle$ MARY, APARTMENT23 $\rangle$: LIVESIN" represents that Mary lives in Apartment23. Axioms $x : C$ and $\langle x, y \rangle : P$ are satisfied by an interpretation $\mathcal{I}$ when $x^{\mathcal{I}} \in C^{\mathcal{I}}$ and $\langle x^{\mathcal{I}}, y^{\mathcal{I}} \rangle \in P^{\mathcal{I}}$, respectively. An interpretation $\mathcal{I}$ satisfies an ABox $\mathcal{A}$ when $\mathcal{I}$ satisfies all the axioms of $\mathcal{A}$. An interpretation $\mathcal{I}$ that satisfies both the TBox $\mathcal{T}$ and the ABox $\mathcal{A}$ is called a *model* of $\langle \mathcal{T}, \mathcal{A} \rangle$.

For completeness, besides A-Box and T-Box statements there exists also R-Box (role box) statements. This type is not supported by OWL 2 ontologies and is usually required in respect of a very expressive description logic.

As already indicated, DLs not only store terminologies and assertions but also allow to reason about them. Typical ABox reasoning tasks are *Consistency*, *Instance Checking*, *Retrieval Problem*, and *Property Fillers* where TBox reasoning is restricted to *Satisfiability* and *Subsumption* [83]. Out of these, we rely on the following ones:

- *Satisfiability:* A class C is satisfiable with respect to a TBox $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}}$ is non-empty.
- *Property Fillers:* Retrieving all the instances in ABox $\mathcal{A}$ that are related to a given individual with respect to a given property.

In this work, we use an ontology to define formally the semantics of ADLs, sensor events, and context data. For that, we use an already existing ontology which was modeled by a knowledge engineer. The ontological reasoning allows to verify the consistency of the ontological model (Satisfiability) and also to derive semantic correlations among activities and sensor events (Property Fillers). Similar to the previous example, the ADL PREPARINGHOTMEAL could be defined based on the definitions of PREPARINGMEAL and PREPARINGCOLDMEAL:

$$\text{PreparingHotMeal} \equiv \text{PreparingMeal} \sqcap \neg\text{PreparingColdMeal}$$

Further, in addition to the common operators (e.g. conjunction) we also consider the following two operators to model certain restrictions:

- *Qualified cardinality restriction.* This restricts the class membership to those instances that are in a given relation with a minimum or maximum number of other individuals of a given class. For instance, the following axiom states that PREPARINGHOTMEAL requires the use of at least one instrument to cook food:

$$\text{PreparingHotMeal} \sqsubseteq \text{Activity} \sqcap \geq 1\, \text{requiresUsageOf}.\text{CookingInstrument}$$

- *Composition of properties.* OWL 2 supports a restricted form of property composition $\circ$. For instance, the following axiom states that if a person is in a given apartment, and she is executing a given ADL, then that ADL is executed in that apartment:

$$\textsc{executesActivity}^{-} \circ \textsc{isInLocation} \sqsubseteq \textsc{actIsExecutedInLocation}$$

Please note that $\textsc{executesActivity}^{-}$ denotes the inverse of $\textsc{executesActivity}$.

2.5 Probabilistic Reasoning

Probabilistic reasoning (also probabilistic logic) combines probability theory (analysis of random phenomena) and deductive logic (reasoning from one or more statements) with the aim of handling uncertainties, imperfection, and contradictory knowledge. Thus, in contrast to the example in the preceding section where we *derived* that Mary has to be a woman, probabilistic reasoning allows to incorporate or handle the possibility that Mary could also be a man. There exists different probabilistic reasoning systems which usually differ by the basic concept and the implementation. On the one hand, a common problem of probabilistic reasoning systems is the computational complexity, i.e., how a reasoner handles and computes probabilistic and logical components. On the other hand, the need to handle many different application scenarios has also lead to many different approaches. In the following, we outline the systems which are probably most known:

- ProbLog2 [84,85] is a probabilistic extension of Prolog which also can calculate both conditional probabilities and most probable explanation (MPE) states.
- RockIt [86] is a maximum a-posteriori (MAP) query engine for statistical relational models
- TheBeast [87] is a software package for statistical relational learning and structured prediction based on Markov logic.
- Tuffy [88] is a highly scalable inference engine for Markov logic networks which use a database backend.

A general distinction has to be made between theoretical concepts and actual implementations, i.e., to which degree a reasoner supports a concept. For instance, while *RockIt* is based on Markov logic networks which in turn generalize First-order logic, the original implementation of *RockIt* does not support numerical constraints. This feature was added later by Huber et al. [89]. Further, *RockIt*, *TheBeast*, and *Tuffy* are all Markov logic based systems [90]; however, Noessner et al. [86] demonstrated that *RockIt* is the most efficient one and outperforms the others in respect of quality.

In contrast, *ProbLog2* is a probabilistic programming language that extends Prolog (a Logic Programming system) where a Prolog program consists of a sequence of Horn clauses (logical formulas). Similar to other reasoners, also *ProbLog2* implements only a subset of the Prolog language[2]. Comparing *Logic programs* with *First-order logic* (both are knowledge representations), one can say that their power of expression overlaps where Horn logic is a common part of both representations. This is another aspect where reasoner can differ.

Overall, each reasoner has to make an assumption about statements where it is not known if it is true or false. Basically, one distinguishes between open- and closed-world assumption. The open-world assumption (OWA) assumes that a statement which is not known to be true or false (based on the considered knowledge base) might be true (i.e. absence of information is interpreted as unknown information). In contrast, the closed-world assumption (CWA) assumes that a statement is false when it is not known to be true or false. As a consequence, the OWA is preferable when the system has incomplete information where in turn the CWA applies when a system has complete information.

Indeed, existing systems may consider both assumptions. For instance, *RockIt* distinguish between *observed* and *hidden* predicates where the former refers to the CWA and the latter to the OWA. This shows that there is no hard border between existing concepts, which in turn is also another reason for a variety of implementations. So far, we only mentioned the tip of the iceberg and for this work, it is out of scope to outline this area in detail. For that reason, we want refer the reader to following literature [91]. However, we believe that the characteristics of a Markov Logic Network (MLN) based system seem to be suitable to reason with sensor data and ADLs; hence, in this work we use *RockIt* [86] together with the numerical constraints extension [89]. For that reason, in following we describe MLN in more detail.

A MLN combines the concepts of a Markov networks (aka. Markov random field) and First-order logic. A Markov network is an undirected probability graph and models a joint distribution of a set of random variables and their conditional dependencies. The overall joint probability distribution is computed as a product of clique potentials (i.e. fully connected subgraphs). More precisely, the probability p is defined as follows:

$$p(s_1, s_2, \ldots, s_n) = \frac{1}{Z} \prod_{c \in C} \theta_c(s_c) \tag{2.9}$$

where Z is a normalization constant which ensures that the distribution sums up to one, C denotes the set of cliques, and θ defines the potential function. For illustration, Figure 2.11 shows a toy example of a Markov network where each node represent a certain sensor. Each sensor might be triggered by a set of ADLs and the edges between the sensors illustrate modeled stochastic dependencies as they might be triggered in respect of the same ADL.

As it is actually unclear which ADL triggered the respective sensors, the question is what the best choice is overall (i.e. the sensors have to *choose* an ADL). Indeed, this is a simplified view because we only want to impart how we intend to use it. For further

[2] `https://problog.readthedocs.io/en/latest/prolog.html`, last access 21.12.2018

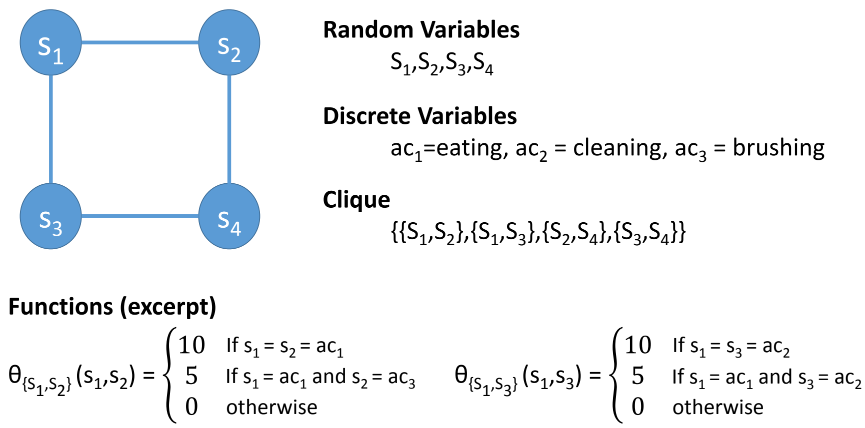


Figure 2.11: Toy example of a Markov network. It depicts four random variables which are illustrated as nodes. The random variables reflect certain sensors that are triggered in respect of certain ADLs (discrete variables). In other words, each of these sensors have to choose between ac_1, ac_2, and ac_3. In this context, the potential functions θ allows to consider knowledge or preferences where the edges model explicit stochastic dependencies, i.e., in this example sensors are linked when the might be triggered in respect of the same ADL.

technical details (e.g. the actual idea of computing cliques), we want to refer the reader to [92].

First-order logic is a formalism for knowledge representation which consists of objects (e.g. Mary), predicates (e.g. is_woman(Mary)), functions (use an object to produce another object), connectives (e.g. $\wedge$ and $\vee$), and quantifier (universal and existential). Compared to traditional propositional logic, the quantifiers and relations allow to formulate more expressive and general sentences. Thus, it allows to define knowledge bases which can be considered as a set of hard constraints. Similar to Markov networks, we do not intend to introduce this formalism but want to give an idea regarding the usage. Referring to our preceding example where Mary is a person (i.e. is_person(Mary)), let us also assume that each person who prepares a meal also eats that meal. The resulting knowledge base would consist of the following predicates and sentences:

$is\_person(x)$	x is a person (e.g. Mary)
$is\_meal(y)$	y is a certain meal
$prepares(x, y)$	x (person) prepares y (meal)
$eats(x, y)$	x (person) eats y (meal)
$\forall x, y(is\_person(x) \wedge is\_meal(y) \wedge prepares(x, y) \Rightarrow eats(x, y))$	if x (person) prepares y (meal), x also eats y

Obviously, this sentence is not always true. This is where MLNs come into play, which soften the constraints of such a knowledge base; hence, in case that a certain formula is violated the world is just more unlikely (and not impossible). Later, we will outline in detail how we construct our knowledge bases. For more technical details, we want refer the reader to [93].

Considering both Markov networks and First-order logic, a Markov logic network can be considered as a template for constructing (large) Markov networks where the general approach is to transfer the idea of Markov networks to First-order logic. Thus, a Markov logic network is a First-order knowledge base with weights attached to constraints.

Technically, a MLN $\mathcal{M}$ is a finite set of pairs $(F_i, w_i), 1 \leq i \leq n$, where each F_i is an axiom (e.g. sentence) in function-free First-order logic and $w_i \in \mathbb{R}$ [90] the corresponding weight. Together with a finite set of constants $X = \{x_1, ..., x_n\}$ it defines the *ground* MLN $\mathcal{M}_{\mathcal{X}}$, i.e., the MLN in which axioms do not contain any free variables. This comprises one binary variable for each grounding of F_i with weight w_i. Hence, a MLN defines a log-linear probability distribution over Herbrand interpretations (possible worlds)

$$P(s) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(s)\right) = \frac{1}{Z} \prod_{c \in C} \theta_c(s_c)^{n_c(s)} \tag{2.10}$$

where $n_i(s)$ is the number of satisfied groundings of F_i in the possible world s and Z is still a normalization constant (cf. Equation 2.9).

As mentioned before, in this work, we use a numerical extension [89,94] which enables to reason on the temporal domain of activities and sensor events and that we denote as $\mathsf{MLN}_{\mathsf{NC}}$. The constraints are predicates of the form $\Theta \bowtie \psi$, where Θ and ψ denote variables, numerical constants, or algebraic expressions (that might contain elementary operators). In this context, the binary operator $\bowtie$ returns a truth value under a particular grounding. More formal, a numerical constraint NC is composed of numerical constants (e.g., elements of $\mathbb{N}$, $\mathbb{I}$), variables, elementary operators or functions $(+, *, -, \div, \%, \sqrt{\ })$, standard relations $(>, <, =, \neq, \geq, \leq)$, and Boolean operators $(\wedge, \vee)$. To be clear, a $\mathsf{MLN}_{\mathsf{NC}}$ is still a set of pairs (FC_i, w_i) where FC_i is a formula in First-order logic that may contain a NC. The following example illustrates how we intent to use it:

Example 1 *Using* $\mathsf{MLN}_{\mathsf{NC}}$ *it is possible to represent the following axiom: two events of "turning on the oven" cannot belong to the same instance of meal preparation if their temporal distance is more than two hours:*

$$\begin{aligned} \{\forall\ se_1,\ se_2,\ ai_1,\ ai_2,\ t_1,\ t_2:\ event(se_1,\ 'oven',\ t_1) \wedge \\ event(se_2, 'oven', t_2) \wedge occursIn(se_1, ai_1) \wedge occursIn(se_2, ai_2) \\ \wedge\ \mathsf{NC}(t_1, t_2) \Rightarrow ai_1 \neq ai_2,\ \mathsf{NC}(t_1, t_2) = |t_1 - t_2| > 120\}. \end{aligned}$$

Based on the resulting MLN_{NC} of sensor events and semantic constraints, we apply Maximum a posteriori inference to derive the most probable activities (most probable world). Maximum a posteriori (MAP) inference is the task of finding the most probable world given some observations also referred to as evidence. Given the observed variables $E = e$, the MAP problem aims to find an assignment of all non-evidence (hidden) variables $X = x$ such that

$$\mathbf{I} = \underset{x}{\operatorname{argmax}}\ P(X = x \mid E = e) \tag{2.11}$$

We denote by $\mathbf{I}$, the assignment x which leads P to be maximal, i.e., a MAP state. In order to compute a MAP state of a MLN, the problem can be formulated as an integer linear program (ILP) using the cutting plane inference algorithm [87]. In respect of $\mathsf{MLN}_{\mathsf{NC}}$, the original cutting planes algorithm [87] was extended to the truth value of numerical predicates on-demand during each CPI iteration [86, 89, 94].

Chapter 3

Related Work

In this chapter, we summarize existing related work to impart existing approaches, research directions, and open issues for both research fields, *Activity Recognition with Wearable Devices* (Section 3.1) and *Activity Recognition within Smart Environments* (Section 3.2). In both cases, we first describe the domain in general and subsequently focus on existing studies which are directly related to our research questions. In addition to this chapter and in respect of our experimental results, we also comprehensively discuss existing works concerning several aspects, issues, and possible future research directions that we identified in the course of this work (see Sections 4.5 and 5.5). Moreover, we also discuss opportunities, advantages, and the necessity of a hybrid solution (see Section 5.5.5).

3.1 Activity Recognition with Wearable Devices

In the following, we focus on physical human activity recognition with wearable devices. First, we briefly outline research elements of interest (Section 3.1.1). Then, we focus on position-aware activity recognition (Section 3.1.2) and personalized cross-subjects activity recognition (see Section 3.1.3) to clarify the state-of-the-art in respect of our research questions (RQ1.x). The following sections were already partly published in [1, 3, 4].

3.1.1 Physical Human Activity Recognition

Almost 15 years ago, Bao et al. [33] published an activity recognition study which is today probably one of the best-known HAR publications. They demonstrated the feasibility of human activity recognition by using five 2D accelerometers and already highlighted the problem of laboratory conditions. Shortly after, Ravi et al. [25] performed similar experiments with a single 3D accelerometer focusing on feature sets and classification techniques to clarify their contribution. Since then, research in Physical Human Activity Recognition spread across various aspects and technical details. This includes the sensor frequency sampling [25, 35], feature selection and computation [95–97], data segmentation [98,99], classification/recognition techniques [37,100–102], sensor positions and orientations [34, 103–105], sensor types [62, 106, 107], subject-dependent and independent approaches [108, 109], and naturally the set of considered activities [18, 23, 39, 110].

The rise of smart-phones gave a new impetus to this domain but also resulted in new issues. Kwapisz et al. [35] recap experiments of preceding works but using the build-in sensors of a smart-phone. As a smart-phone is usually located in a pocket, it also moves slightly while the user is moving. Their results show the reliability of a smart-phone but also illustrate (frequent) confusion between certain activities, even when considering windows of 10 seconds. Another issue is the energy consumption, as a smart-phone is an object that is frequently used and should be permanently available, the energy consumption of the activity recognition application needs to be minor. Casale et al. [111] focus on feature sets which are competitive from computational point of view. In this context, they clarify the importance of the considered features for the respective classifier. Another aspect is the on-body location of a smart-phone. Mannini et al. [112] highlight that using only sensor data that were record close to the hip may underestimate the overall expenditure on activities but also that walking with a big bag or a small cup results in different sensor signals. In general, they propose a sensor placed at the wrist or ankle to handle these issues but they also highlight a number of weaknesses. Indeed, it can be assumed that a combination is more productive. This in turn leads to the question how to fuse different sensor streams. Shoaib et al. [113] investigates combinations of different sensor types considering different scenarios. They apply late fusion, i.e., they compute features for each sensor stream independently. They conclude that the impact of a certain sensor type or respective combinations depends on the scenario.

Apart from that, several publications frequently summarize and highlight the rapid development [37, 114, 115]. Indeed, activity recognition related research has become a regular topic in international conferences including *AAAI*, *CVPR*, *IJCAI*, *NIPS*, *PERVASIVE*, *UbiComp*, *PerCom*, *ISWC*, *ICAPS* and *AMI* [114]. All this shows the scope of the research domain and that it is not reasonable that we dive into each aspect. Therefore, in the following, we focus on studies that are directly related to our introduced research questions, i.e., who deal with position-aware activity recognition and personalized cross-subject recognition models. Beyond that, we also examine relevant works in respect of our experimental results and subsequent discussions.

3.1.2 Position-Aware Activity Recognition

As previously described, several researchers have already investigated activity recognition independent of the device position [116]. However, many studies state that the device position information increases the accuracy of an activity recognition algorithm but the opinion regarding the impact of this information on the respective results differs significantly [18, 34, 117]. This difference is due to varying sets of positions and activities considered in the respective studies. Indeed, so far nobody considered all relevant body positions and common physical activities in a single study. Therefore, it is still unclear how accurate each relevant position can be detected regarding different physical activities.

The on-body localization problem of wearable devices plays an important role because it can help to improve the accuracy of activity recognition, to optimize the energy consumption of a device, or to increase the precision of observing the environment. This is a consequence of the results of related studies. They investigated the influence of the on-body position to determine optimal sensor placement in context of activity recognition [34,117,118]. They show that there are seven body locations, which behave differently in respect of the same activity. In particular, forearm, head, shin, thigh, upper arm and waist/chest. Dividing these body parts (e.g., head) into smaller regions does not improve the accuracy [118]. Further studies have shown that the optimal sensor placement depends on the activity to be recognized [34]. As a result, the benefit of the position information and the feasibility to derive the device positions by an accelerometer are concluded; however, it is not clear to what extent.

So far, the device localization problem was only addressed by a couple of researchers. Kunze et al. [103] published one of the first approaches where they tried to detect if the user is walking and subsequently to map specific patterns of sensor readings to derive the current device position. However, this approach is limited due to the small set of selected positions and the fact that position changes are not recognized if the user does not walk. Recently, researchers investigated also the possibility to derive the positions hand, bag, and pocket from different physical activities [18]. They state that the effect of the location information on the accuracy of the activity recognition depends on the performed activity.

A number of studies also tried to develop a location independent activity recognition approach by learning a generic classification model for all positions [116] but several subsequent studies state that a position-specific activity recognition performs always better than a position-independent activity recognition [18, 34, 117].

While these studies focus on on-body position detection with an accelerometer, several researchers also examined the possibilities to detect if the smart-phone is located in- or out-pocket [119], in a bag [18, 120], or still worn by the same person [121]. They also used other sensors such as a microphone, light, or proximity sensor. They highlight that an accurate detection is possible but also point out that it is difficult to control the environment regarding brightness or sound level, which has to be considered as the crucial problem.

3.1.3 Personalized Cross-Subjects Activity Recognition

Subject-specific activity recognition has been extensively investigated by many researchers [33, 35, 101]. They achieved reliable recognition rates in many different scenarios but required for each subject a labeled training set. Further, changes in the user's motion patterns are often not considered by the proposed methods, which leads to a worse recognition rate over time.

As a first approach to reduce the need of labeled data, researchers have investigated cross-subjects approaches. Especially, the leave-one-subject-out approach was evaluated comprehensively and researchers state that it performs significantly worse compared to a subject-specific classification model [23, 106, 122]. This even holds if several acceleration sensors are considered simultaneously [106]. The researchers conclude that this is due to differences in the physical characteristics of the considered subjects, e.g., fitness level, gender, and body structure. Indeed, researchers hypothesize that these kinds of characteristics could be reliable indicators to identify subjects with similar acceleration patterns [108, 122]. So far, this assumption was only considered in few works. Maekawa et al. [123] applied this concept successfully and they conclude that a minimum number of subjects is required. However, the authors used five acceleration sensors and also considered complex activities (e.g., play pingpong) which makes it difficult to interpret the aggregated results. Besides, in some works models were trained on one person and used on another without considering any characteristics [25, 124, 125]. They state that such a model often cannot yield accurate results if it is used on different subjects and that a personalization is required. In respect of our research questions, we focus on this hypothesis but also investigate cross-subjects approaches concerning their performance in context of all relevant on-body device positions and combinations.

Instead of using labeled training data across people, several researchers also investigated semi-supervised approaches, e.g., active learning, to reduce the labeling effort [126–128]. These works aim at extracting the most informative unlabeled samples to minimize the user interaction. By using active learning, the user could be queried regarding these samples to gain new knowledge. Their results show that active learning does improve the learning performance and that it is possible to achieve comparable recognition rates with respect to a supervised approach [126]. In this context, the most informative unlabeled samples could be identified by interpreting the classifiers confidence values [129]. However, this approach still requires a small, initial labeled dataset in respect of the target user.

Indeed, using labeled data across subjects and interactively querying the user (active learning) do not exclude each other but are complementary. Labeled data could be used across subjects to build a base model that could be personalized by knowledge that was gathered by querying the user. So far, personalization of an existing activity recognition model was realized by updating parameter of an existing model [39, 130], or incremental learning [131–133]. In this context, researchers evaluated neural network [132, 134], support vector machine [135], and fuzzy rule [54, 136] based approaches and even if the results of these works are difficult to compare due to the different setups, the results show that the concept of personalization is feasible. Besides, to gather additional information from unlabeled sensor data, researchers also applied successfully the concept of co-training [38, 126].

So far, nobody combined all of these techniques or aspects where in addition especially the mentioned personalization approaches have limitations. Concerning parameter adaption, the structure of the model is almost fixed where incremental learning has to keep all data available and usually does not distinguish between newer and older information. Indeed, some of these works also apply re-training to process new gathered data, which is often unfeasible. In this context, the influence and performance concerning the users' effort that goes along with active learning or the relation concerning the number of uncertain samples, queries, and achieved improvement is also unclear.

3.2 Activity Recognition within Smart Environments

In the following, we focus on recognizing ADLs in a smart environment. First, we summarize limitations and drawbacks of existing approaches (Section 3.2.1). Then, we focus on online recognition of interleaved ADLs (Section 3.2.2) and collaborative and active learning in a smart-environment (Section 3.2.3) to clarify the state-of-the-art in respect of our research questions (RQ2.x). The following sections were already partly published in [2, 5, 8].

3.2.1 Recognizing Activities of Daily Living

In general, human activity recognition techniques in pervasive computing can be broadly classified in two categories, namely learning-based methods and specification-based methods [49].

Learning-based methods rely on supervised learning algorithms and consider a training set of labeled sensor data to build the recognition model. As one might expect, this includes physical human activity recognition systems which rely on wearable sensors such as accelerometers [33, 37] or those that acquire the surrounding area (e.g., microphones) [137, 138]. Focusing on complex activities, observations regarding the user's surrounding area (in particular, objects' use), possibly coupled with wearable sensors, are the basis of other activity recognition systems [14, 139]. Indeed, these studies use the basic idea of a hybrid solution but in a limited way. However, since training data is hard to acquire in realistic environments and may violate the individuals' privacy [140], systems relying on supervised learning are prone to serious scalability issues the more activities and the more context data are considered. Moreover, datasets of complex ADLs are strongly coupled to the environment in which they are acquired (i.e., the home environment and the sensors setup), and to the mode of execution of the specific individual. Hence, the portability of activity datasets in different environments but also suitable transfer learning methods for activity models are open issues [41, 141].

Specification-based methods rely on knowledge-based definitions of the characteristics and semantics of complex activities, i.e., complex activities are defined in terms of their simpler components. Sequences of simple actions, recognized by certain sensors, are matched to activity definitions to identify the occurred activity. Those definitions are usually expressed through logical axioms, rules, or description logics [44–46, 142]. However, complex activities are characterized by large variability of execution. In order to cope with that issue, other works investigated the use of less rigid formalisms to define ADLs. Helaoui et al. [47] used probabilistic description logics to define a multi-level ontology of domestic activities but as most approaches, they require significant knowledge engineering efforts, and are hardly scalable to the definition of a comprehensive set of ADLs in different contexts. Beyond that, ontological reasoning has also been proposed to perform dynamic segmentation of sensor data [99, 143, 144] or to refine the output of supervised learning methods [145]. Further, defeasible reasoning has been adopted to enhance existing sequential activity recognition systems by detecting interleaved activities and handling inconsistent or conflicting information [146]. However, those works rely on rigid assumptions about the simpler constituents of activities [142]. Hence, while the specification-based approach is effective for activities characterized by a few typical execution patterns, it is hardly scalable to the comprehensive specification of complex ADLs in different contexts.

This is where we investigate if the recognition of complex ADLs through semantic reasoning is feasible to overcome the requirement of a large expensive labeled ADLs dataset. Using the introduced ontological reasoner and a suitable knowledge base may allow to identify general semantic correlations between the smart-home infrastructure and performed activities. Moreover, when training data is available, we can also exploit it to mine low-level dependencies between them. Indeed, the combination of specification-based and probabilistic approaches is not new and has been investigated in other fields of Artificial Intelligence [147]. However, in contrast to most existing techniques, we target the recognition of interleaved ADLs explicitly by considering this aspect in our MLN_{NC} model. This enables us to assign sensor events to overlapping activity instances. This reflects situations where the actual ADL is briefly interrupted by another activity (e.g., someone stopped eating to take medicine).

Considering unsupervised learning techniques which avoid manual specification, researchers usually build activity models by mining various sources (e.g., Web resources, or unlabeled datasets of activities). A first attempt in this sense was due to Perkowitz et al. [148] which in turn was refined in later works [149–151]. Those methods analyze textual descriptions of activities mined from the Web in order to obtain correlations among activities and objects used for their execution. Those correlations are used to recognize the executed activity based on the observed sequence of used objects. That approach has been recently extended to exploit visual cues extracted from the Web, such as images and videos [53]. However, it is questionable whether object-activity correlations are sufficient to recognize complex ADLs. As an example for mining unlabeled data, Rashidi et al. [152] introduce an automated approach to activity tracking that identifies frequent patterns that naturally occur in an individual's routine. An unsupervised method that is close to our approach has been proposed by Ye et al. [153]. In particular, they introduce a knowledge-based method which computes similarities among pairs of sensor events based on their temporal, spatial, and usage dimensions. In this context, objects similarity is used to segment sensor event traces that should represent the execution pattern of a single activity. Subsequently, sequential pattern mining is used to identify frequent sequences of sensor events that typically appear during an activity. Exploiting an ontology of activities and objects, each frequent sequence is associated to one or more activities, according to the objects that triggered the sensor events in the sequence. Finally, sequences are refined by a clustering algorithm, and refined sequences are used for activity recognition. With respect to that work, we focus on an approach that is very independent from the data and can also handle interleaved activities.

3.2.2 Online Recognition of Interleaved ADLs

Several works considered the challenging issue of segmenting temporal sequences of sensor data to recognize accurately the boundaries (i.e., start- and end-time) of activity instances in real-time.

First, Yin et al. [154] propose to segment and recognize complex activities based on the user's location trace. Their method relies on a signal-strength distribution at each sampled location, which determines the probability distribution of current activities. In order to cope with variability and imprecision of signal-strength data, they represent the motion pattern as a linear dynamic system, and adopt a transition matrix among motion patterns to model the nonlinear dynamics of the stochastic process of activities. Activity recognition and segmentation is achieved by applying an approximate Viterbi inference algorithm. While this work only considers the location of the user, it also models the hidden states with a first-order Markov chain.

Second, Palmes et al. [151] propose two unsupervised segmentation methods (*MaxGap* and *MaxGain*) based on correlations among used objects and activities. Those correlations are extracted from the textual content of web pages and computed by the well-known *tf-idf* function. The extracted correlations enable to estimate the discriminative power of an object towards activities. Objects with high discriminative power are named *key objects* where activities are recognized based on the observation of their key objects' usage. In the *MaxGap* algorithm, the boundary between two activities is predicted at the time of usage of the most discriminative object between the two. In the *MaxGain* algorithm, the boundary is the one that maximizes the sum of correlation values between the two activities and the used objects. A drawback of their approach is the assumption that each activity has a unique key object. That assumption may be unrealistic in several scenarios.

Third, Okeyo et al. [99] propose the use of different heuristics to segment activities in a knowledge-based framework. Those heuristics consider the activity duration and semantic features to shrink and expand a dynamic time window of activities. However, their approach leaks on handling activities that occur in patterns, as they did not consider temporal information.

Fourth, Wan et al. [155] propose a supervised segmentation approach based on the correlation between consecutive sensor events, and on the time distance between them. Limitations are that they consider only relations between pairs of sensor events and naturally that their technique requires a labeled dataset. Besides, similar to Palmes et al. [151], most considered activities are correlated with a unique sensor event.

Fifth, Aminikhanghahi and Cook [156] propose to segment a stream of sensor events using an unsupervised change point detection algorithm, and subsequently to recognize each segment's activity using a supervised learning approach. However, the presented approach mainly tries to recognize activity transitions and requires a pre-segmentation with a fixed window length. Further, the corresponding results are unclear concerning the segmentation quality.

Last, Triboan et al. [157] present a semantic technique for online segmentation and activity recognition. In that work, segmentation and recognition rely on ontological and rule-based definitions of activities. Unfortunately, the authors do not provide any statistical evaluation in respect of performance, feasibility, or quality.

We want to overcome most of these limitations by considering correlations and aspects among activities and sensor events and also by using MLNs which allow to model arbitrarily complex (temporal) constraints. On the one hand, this allows to focus on multiple types of sensor data while on the other hand our method does not require a labeled training dataset. Further, we not only resize the time window of ADLs, but we also re-arrange segments to cope with interleaved activities. In this context, we consider all preceding segments to optimize the segmentation process. Besides, we expect that this concept is flexible enough to be easily adaptable to different environments and execution modalities.

3.2.3 Collaborative and Active Learning in a Smart-Environment

Semi-supervised learning methods use unlabeled data to improve the model computed through a training set. In this context, active learning has the intention to use those unlabeled data to query people with the purpose of reducing the level of supervision. Essentially, two questions go along with such an approach. First, which data in particular should be considered for querying and second, who should be asked.

A number of researchers have focused on the first question. In particular, Stikic et al. [126] investigate the use of active learning, with the objective of identifying the most informative sequences of sensor events for which to query the user. A sequence is considered informative either when the confidence of the classifier about its predicted activity class is low, or when two classifiers disagree about its class. They conclude that it is possible to achieve comparable or sometimes even higher accuracy than the fully supervised approaches with less labeling efforts. In contrast, Ho et al. [158] propose to use active learning especially for understanding changes in the home environment to adapt subsequently the recognition model. In that work, an entropy-based measurement is used to query the most informative sequences of sensor events to update a Dynamic Bayesian Network. Further, Zhao et al. [159] proposes three more techniques to choose the most informative data points for which to query the user. These methods are based on (i) low confidence for the most probable activity class, (ii) small difference between the confidence of the most and second most probable class, or (iii) high entropy among the probability of classes. However, their experimental results in smart-home settings show that these three methods achieve similar accuracy.

In respect of the second question, significantly less researchers focused specifically on reducing the load on the individual. An active learning method to refine iteratively the annotations of a video provided by crowdsourcing services (like Mechanical Turk) is presented in [159]. That method relies on confidence scores about the annotation where annotations with low confidence are re-submitted to the crowdsourcing service for revision. A similar approach is proposed by Lasecki et al. [160]. In their work, privacy of individuals depicted in videos is protected by automatically identifying people and veil them by coloring their silhouette. Further, the work presented in [161] proposes strategies to select the most appropriate annotators in a crowdsourcing framework for active learning of ADLs. To achieve a high information gain with a few questions, Hoque et al. [162] used data mining methods to cluster sequences of sensor events, such that each cluster represents an activity class. Subsequently, the resident is asked to provide the actual class for each cluster.

Apart from active learning, other works propose transfer-learning methods to reuse activity datasets acquired in different environments [141]. However, effective portability of activity datasets is challenging, since datasets of complex ADLs are strongly coupled to the environment in which they are acquired and to the mode of execution of the individual [41]. A related issue is how to adapt dynamically the recognition system to changes in the sensor infrastructure. With this regard, a technique was proposed to update the model of a supervised machine learning algorithm with features of newly discovered sensors [50].

In contrast to these works, we focus on a *collaborative* active learning method which allows to share the burden of providing ADLs labels among a community of residents. The idea is to exploit users' feedback across different smart-environments to assign a certain semantics to sets of sensor events. A critical part is how to consider different home characteristics but also the residents themselves. For that, we focus on a similarity measure between the context of the target environment (characteristics of home and resident) and the one of the environment from which the label is acquired. We expect that this approach also enables to integrate easily information about new sensors.

Chapter 4

Activity Recognition with Wearable Devices

In this chapter, we focus on the introduced open issues in respect of physical activity recognition with wearable devices (see Section 1.3) and present related approaches, solutions, experiments, and discussions.

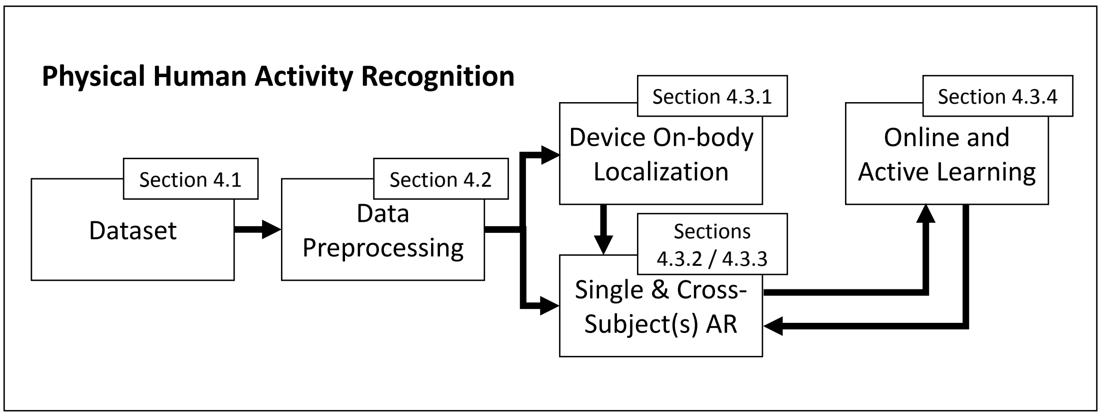


Figure 4.1: Physical Human Activity Recognition with Wearable Devices

For that purpose, first we explain the data gathering process regarding the required dataset (Section 4.1, published in [1]). Subsequently, we introduce the preprocessing techniques for the data handling but also for improving the quality concerning irrelevant and redundant information (Section 4.2, published in [1]). Then, as a first step for improving physical activity recognition, we present an approach that addresses the device on-body localization problem using only acceleration data (Section 4.3.1, published in [1]). Based on this, we introduce a position-aware activity recognition approach for clarifying the influence of the position information (Section 4.3.2, published in [1,3]). As this approach focuses only on single-subject based models, we also investigated the possibility of cross-subjects based recognition models to overcome the data gathering effort (Section 4.3.3, published in [3,4]). Finally, we present a solution to evolve a physical activity recognition model over time, i.e., to adapt it to changes in behavior of the user (Section 4.3.4, published in [4]). Figure 4.1 illustrates the structure and content of this chapter. Please see Appendix A for further details regarding the contribution of the individual authors.

4.1 Physical Human Activities Dataset

To answer our initial research questions and to investigate the related issues, it is necessary to create a new dataset, as the existing ones do not full-fill our requirements. In particular, this includes especially the lack of sensor data of each relevant on-body device position in respect of the considered physical activities but also the lack of transparency of the data recording sessions. In the following, we introduce our self-developed data collection tool for smart-devices and subsequently we describe the recording session and the dataset. The following subsections belong to the publication [1].

4.1.1 Sensor Data Collector

For a better understanding of the sensors but also to verify the feasibility of our approaches, we decided to develop an Android application that allows to record and label all types of sensors which are available in smart devices today. We choose Android instead of iOS as development platform because it allows to access the raw sensor readings directly, i.e., without intermediate filters.

Figure 4.2: Sensor Data Collector. The framework consists of a wear (1) and hand (2) app which allows to record each sensor that is available and it provides labeling and visualization functions.

Overall, our application (app for short) is two-parted and consists of a *Wear* (1) and *Hand* (2) app (see Figure 4.2) which interact via Bluetooth. The *Hand* app is the central control unit, works standalone, and runs on *Android* (e.g. smart-phones) while the *Wear* app was designed for *Android Wear* (e.g. smart-watches) and can be considered as an extension. In contrast to the *Hand* app, the *Wear* app offers only a subset of functionality which include sensor recordings, labeling of readings, and streaming of this data directly to the main device for live plotting, analyzing, and storing (see Figure 4.3). In addition, the *Hand* app provides the possibility to specify which sensor types should be recorded (simultaneously) and with which frequency, allows to export the recorded data in various formats, and enables to correct previous set labels.

A common usage scenario would be to mount a smart-phone (*Hand*) to any on-body position (e.g. pocket) and start the sensor recording while the smart-watch (*Wear*) could be used to adapt the current label to current situation without interacting with the smart-

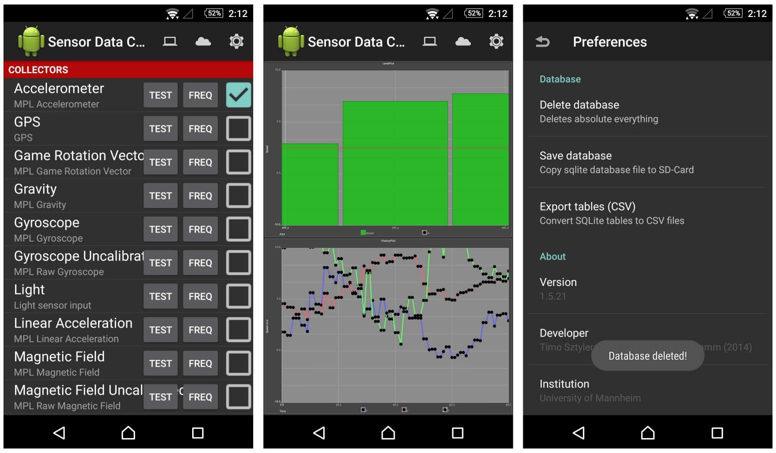


Figure 4.3: Hand App Interface. The screens show the features that are only provided by the *Hand* app: sensor management (left), plotting (middle), data export (right)

phone, i.e., without producing noise in respect of the sensor readings. The binary[1] and the source code[2] of this application are publicly available.

4.1.2 Data Gathering

We create our dataset[3] with the introduced *Sensor Data Collector* where we record the activities climbing stairs down (ac_1) and up (ac_2), jumping (ac_3), lying (ac_4), standing (ac_5), sitting (ac_6), running/jogging (ac_7), and walking (ac_8) of fifteen subjects (age 31.9 ± 12.4, height 173.1 ± 6.9, weight 74.1 ± 13.8, seven females). For each activity, we observed simultaneously the body positions chest (op_1), forearm (op_2), head (op_3), shin (op_4), thigh (op_5), upper arm (op_6), and waist (op_7). Each subject performed each activity roughly 10 minutes except for jumping due to the physical exertion ($\sim$1.7 minutes). Overall, we recorded for each position and axes 1065 minutes where the data is equally distributed concerning male and female. Table 4.1 summarize in detail the characteristics of our dataset. To the best of our knowledge, the result is the most complete, realistic, and transparent dataset for on-body position detection that is currently available (September 2015).

The required data was collected using customary smart-phones and a smart-watch ("Samsung Galaxy S4" and "LG G Watch R") which were attached to the mentioned positions (see Figure 4.4). The devices were synchronized with the time service of the network provider and the sensors were sensed with a sampling rate of 50 Hz where the data was stored on a local SD card. The sampling rate was chosen with consideration of battery life as well as with reference to previous studies [25, 35].

To attach the devices to the relevant body positions, common objects and clothes were used such as a sport armband case, trouser pocket, shirt pocket, or the bra. There was no further fixation of the device to resemble closely their use in everyday life. In case

[1] https://play.google.com/store/apps/details?id=de.unima.ar.collector
[2] https://github.com/sztyler/sensordatacollector
[3] https://sensor.informatik.uni-mannheim.de/

Figure 4.4: Sensor placement. The subject wears the wearable devices on the head, chest, upper arm, waist, forearm, thigh, and shin (top down).

Table 4.1: Dataset. Length of the recording for each activity and each device position in minutes.

Activity	**Total [min]**	**Average [min]**	**Female / Male [min]**
climbing down (ac_1)	123.55	8.24	54.88 / 68.67
climbing up (ac_2)	148.43	9.89	70.33 / 78.10
jumping (ac_3)	24.93	1.66	11.52 / 13.41
lying (ac_4)	157.14	10.47	73.25 / 83.89
standing (ac_5)	154.18	10.27	72.27 / 81.91
sitting (ac_6)	156.65	10.44	73.57 / 83.08
running (ac_7)	140.69	9.37	71.52 / 69.17
walking (ac_8)	159.45	10.63	74.41 / 85.04
all	1065.02	8.88	501.75 / 563.27

of the head we used a belt to avoid that the subject had to hold this device during the performance of the activities. This simulates that the subject phones during the activities.

The data collection took place under realistic conditions, i.e., the subjects walked through the city, jogged in a forest, or climbed up the stairs of a guard tower of an old castle. The order of the activities was left to the subjects but they were instructed to stand idle for a few seconds before and after an activity was performed. Concerning the activities, there were no instructions. It was up to the subject, e.g., how fast they wanted to walk or how they wanted to sit. In this context, typically the subjects used their smart-phone (that was not used for recording), talked with somebody else, or were eating and drinking something while they were standing or sitting.

Each movement was recorded by a video camera (third-person view) to facilitate the usage of our dataset also by other people. Our dataset is available[3] and covers the data of accelerometer, GPS, gyroscope, light sensor, magnetometer, and sound level sensor. Besides, we also provide a detailed description of each subject including images of the attached devices and a short report.

Compared to the well-known datasets *OPPORTUNITY* [163] and *COSAR* [145], we did not focus on activities of daily living but physical activities. Indeed, it would be possible to derive the physical activities from the activities of daily living and both datasets also cover acceleration data from on-body devices, however, several aspects and activities are not covered. On the one hand, *OPPORTUNITY* covers several different on-body positions but provides only one single dynamic activity (walking) where on the other hand the *COSAR* dataset covers several different physical activities but provides only acceleration data for two on-body positions. Besides, both datasets cover significant fewer subjects (four and six) which are too few to analyze, e.g., physical characteristics or certain groups of people.

4.2 Data Preprocessing

The data preprocessing step consists of the segmentation of the sensor data and the computation of features based on the segmented data. Here, one aims to compute features for each segment which are characteristic for a performed activity within the respective time interval. In the following, we will go into detail and describe the corresponding methods and techniques. The following subsections belong to the publication [1].

4.2.1 Window Segmentation Techniques

The segmentation of sensor data or a sensor stream aims to isolate individual actions of an activity within segments also called windows. Therefore, a window can be considered as a certain time interval with a start and stop timestamp that comprises sensor readings which were recorded during that time interval. Such a window allows to compute characteristic features (e.g., turning points) based on raw sensor values that represent the corresponding

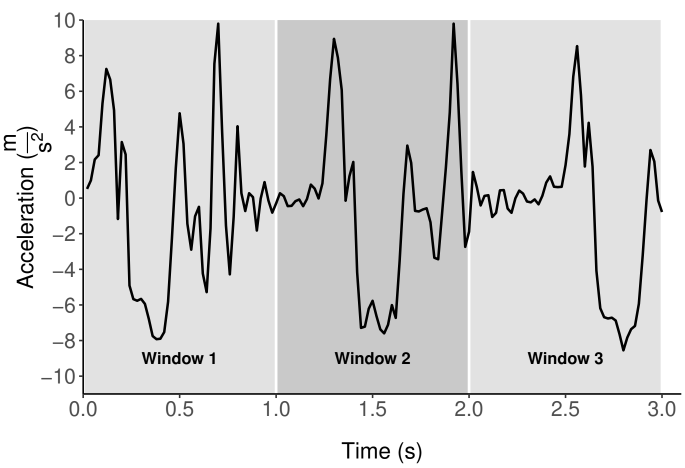


Figure 4.5: Acceleration signal of a smart-device that is attached to a human body while walking. It depicts a simple static windowing approach to capture the repeating pattern.

action. The goal is to recognize the initial activity based on the computed features. Computing features instead of using raw sensor data directly to recognize an activity helps for one thing to reduce noise, as the sensor signal is fluctuant, for another thing it ensures to consider the distinctive characteristics.

For instance, Figure 4.5 illustrates acceleration data that was recorded while walking and it clearly depicts a repeating pattern. This pattern can be considered as the individual footsteps and we aim to encapsulate each of these footsteps in separate segments. Please note that this figure is just a simple example, i.e., patterns are not always this obvious and could be also spread across several dimensions.

As the segmentation of the data should be performed automatically, one has to rely on rules or requirements that have to be fulfilled to determine the start and stop time of a window. Indeed, there are different approaches that can be basically grouped by static and dynamic windowing (see Figure 4.6). In case of a static window, each window has the same predefined length where a dynamic window varies in length. The former is especially preferable when the pattern is almost constant and repeating (cf. Figure 4.5) but also in case when there are no usable signals or characteristics to make a decision. In contrast, a dynamic window approach can be used by relying on turning points, variations, outlier, or extremes. This is useful if a certain activity should be recognized as in case of a acceleration-based fall detection system [7].

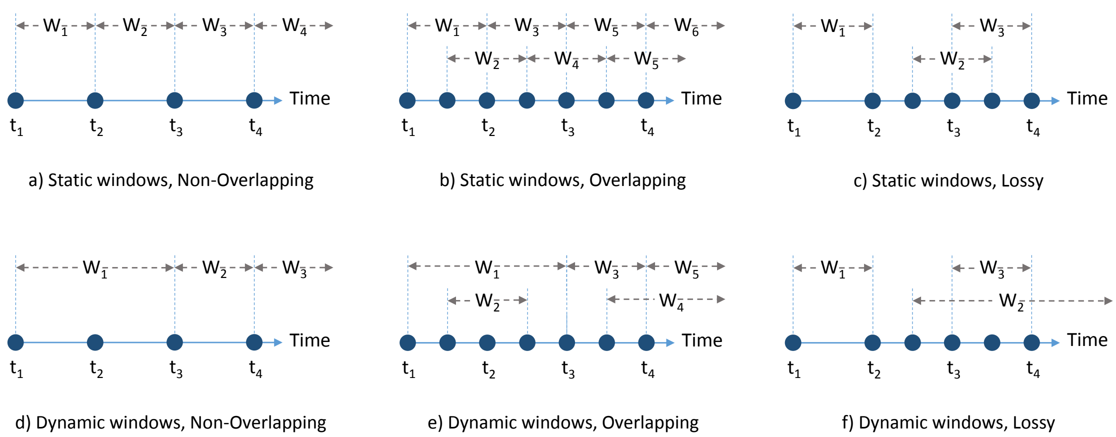


Figure 4.6: Static and dynamic windowing approaches for activity recognition. The most suitable or applicable approach depends on the data stream and target activity. For instance, an overlapping approach is suitable for capturing transitions between activities where a dynamic approach is preferable if it is possible to identify transitions or certain characteristics based on the data stream.

In addition, windows can be concatenated (cf. Figure 4.6 a) and d)), can overlap (cf. Figure 4.6 b) and e)) or can be treated independently (cf. Figure 4.6 c) and f)). The latter can be considered as a special case that is used to interpret certain signals, i.e., a window could be create just around a spike. In context of physical activity recognition, a static overlapping window approach is preferable as movements like walking or running are almost constant in respect of duration and execution but there could be slight variations but also transitions between different activities. We will return to dynamic windows when we consider ADLs.

4.2.2 Feature Extraction

The essential idea behind generating features from time depended data streams is to segment the recorded data into windows and compute a feature vector for each window. Preceding studies in context of physical activity recognition already examined different settings regarding the window size and meaningful features [108]. They state that overlapping windows are more suitable because they can handle transitions more accurately. Further, the window size depends on the kind of activities which should be recognized. In our context, most of the existing studies considered a size between one and three seconds [95,105,116]. However, so far there is no agreed set of features. Indeed, a comparison of the different but overlapping feature sets of previous studies is difficult due to the different settings and goals of the studies. Nevertheless, some researchers have compared different groups of features and stated especially that frequency-based features improve the accuracy of the recognition [95]. Based on these results, we use windows which overlap by half and have a length of one second. Further, we consider the most common time- and frequency-based features that were used in previous work (see Table 4.2).

Table 4.2: Summary of considered feature methods.

Domain	Methods
Time	correlation coefficient (Pearson), entropy (Shannon), gravity (roll, pitch), mean, mean absolute deviation, interquartile range (type R-5), kurtosis, median, standard deviation, variance
Frequency	energy (Fourier, Parseval), entropy (Fourier, Shannon), DC mean (Fourier)

Time-based features are directly computed from the recorded sensor data. As usually the orientation of a smart-device can change and so the orientation of the axes, it is important to compute orientation-independent features for recognizing certain patterns as it is unfeasible to consider each possible orientation. However, the device orientation may provide usual information but it has to be used carefully. For instance, we also computed gravity-based features that provide information of the device orientation in the form of angles. In detail, the gravity component can be extracted from the recorded acceleration data by applying a *low-pass filter*[4] which separates the linear acceleration and gravitational force to derive the gravity vectors. These vectors allow to determine the orientation of the device by computing the angles between them also known as *roll* and *pitch* (see Figure 4.7). The *azimuth* angle, however, cannot be calculated because the direction of north is required (magnetometer, see Section 2.2.1.3). This means that it is not possible to derive from an accelerometer if the device is back-to-front. Further,

[4]A low-pass filter passes values which have a lower frequency as the specified *cutoff frequency* and attenuates values that have a higher frequency.

we only consider absolute value of the acceleration so that we do not distinguish if the device is upside down. Hence, we consider these four cases as the same position. To be more flexible and avoid overfitting, we also transform the roll and pitch angles in one of sixteen predefined discretized orientations. Besides, we analyze gravity-based features only in respect of on-body position detection.

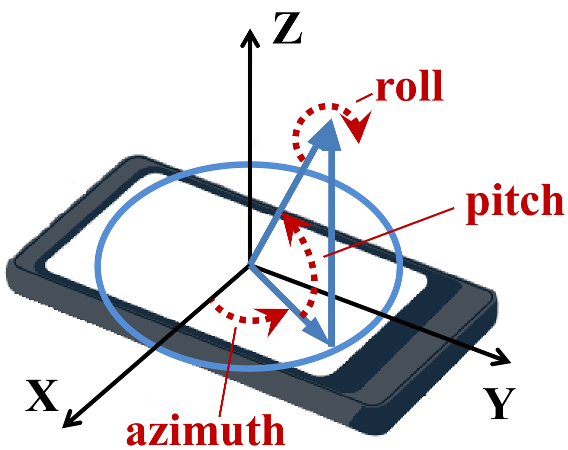


Figure 4.7: The coordinate system is defined in reference to the screen. The acceleration of the device is measured along the axes. The gravity enables to compute the angle between the axes to determine the orientation (roll, pitch). To calculate azimuth, the direction of north is required.

Frequency-based features are computed based on the values that result from the Discrete Fourier Transformation (DFT). Meaning, this technique transforms data from the time-domain into the frequency-domain but also vice versa (Fourier synthesis); hence, the transformation is lossless but enables to analyze the same data from a different perspective. In this context, the time domain values represent, e.g., acceleration dependent on time where the frequency domain represents the magnitude dependent on frequency (hertz). A drawback of the transformation is the runtime complexity which is usually $O(n^2)$ and results from a matrix multiplication as the input data has to be mapped to complex numbers. This might be a problem for real-time application scenarios. However, if the number of input values is 2^x then the runtime complexity can be reduced to $O(n * log(n))$. This case is also known as *Fast Fourier Transformation* (Radix-2-Algo). Further, if the input data consists only of real numbers then only the first half has to be computed, as the result is symmetric. Both requirements can be full-field in our scenario. Overall, this allows us to compute, for instance, the *Energy* that was required to perform an acceleration in a certain time span. The Fourier transformation can be applied with different scaling factors. We use the *JTransforms*[5] implementation which scales by one.

The feature extraction process was performed with a self-developed framework that computes all mentioned features. The framework is available[6] and allows to specify the mentioned settings. As a result, the framework returns a list of feature vectors which are

[5] `https://github.com/wendykierp/JTransforms`
[6] `https://github.com/sztyler/sensorfeatureextraction`

in the following further processed. A detailed description of the implemented features is attached (see Appendix B). Furthermore, as more and more researches propose an *Autoencoder* for feature generation, we want to emphasize that the size of our dataset is inappropriate.

4.3 Methods

The computed windows and the corresponding features are the input of the methods presented below. In particular, the recognition of the device on-body position (see Section 4.3) and based on these results single-subject (see Section 4.3.2) and cross-subjects (see Section 4.3.3) based physical activity recognition models. Finally, we present an approach for adapting those models at runtime to the user's behavior by online and active machine learning. The following subsections belong to the publications [1, 3, 4].

4.3.1 Device On-body Localization

We treat position detection as a multi-class classification problem with target classes being head, upper arm, forearm, chest, waist, thigh, and shin that correspond to the relevant position according to Vahdatpour et al. [118].

In initial experiments, we observed a major problem when trying to distinguish between different device positions while considering all performed physical activities. More precisely, data of the activities lying, standing, and sitting frequently leads to misclassification of device positions. This is caused by the fact that in context of these three activities the human body only has a slight acceleration so that the computed feature vectors are not easily distinguishable. To address this problem, we distinguish between static (standing, sitting, lying) and dynamic (climbing up/down, jumping, running, walking) activities and consider these two groups in the following as two types of activity-levels. This enables to consider different features sets. Hence, we train a classifier that distinguishes between static and dynamic activities that is used as a first step in the position detection process. A similar distinction has been made in [164] to improve the accuracy of activity recognition.

The prior distinction between static and dynamic activities (and thus the possibility to use different feature sets) enables especially to use gravity-based features in context of static activities. Figure 4.8 illustrates the changes of the device orientation that result from the different postures. In contrast, the dynamic activities are usually performed in an upright position (cf. standing).

We trained both models using stratified sampling combined with 10-fold cross validation to ensure that all folds cover the same ratio of classes. Further, to make the result more stable, we performed 10 runs where each time the dataset was randomized and the 10-folds were recreated. The classifiers were trained and evaluated for each subject individually (single-subject).

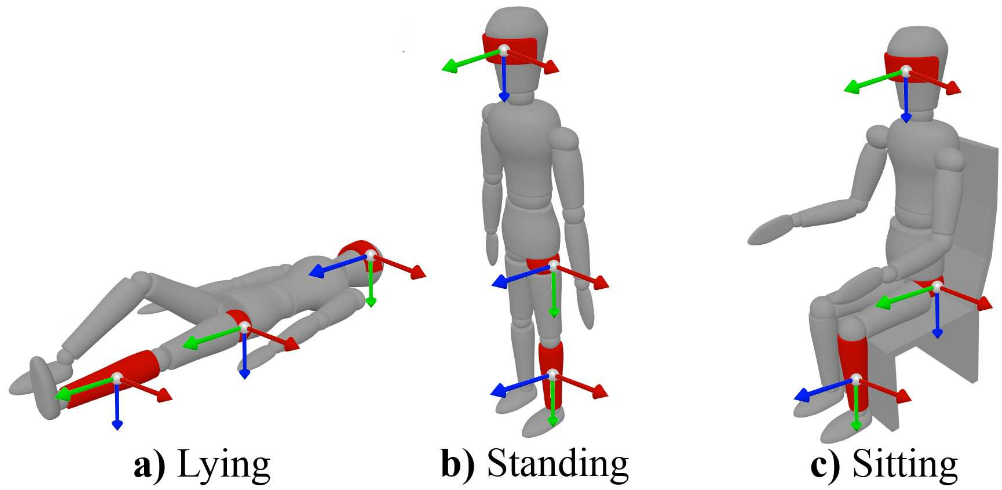


Figure 4.8: The change of the orientation of a device when changing between lying, standing, and sitting for the on-body positions shin, thigh, and head.

4.3.2 Single-Subject Position-Aware Activity Recognition

In the activity recognition phase, we aim to detect the activities climbing stairs up and down, jumping, lying, running, sitting, standing, and walking. In this context, we evaluate the impact of the information of the device position. For this purpose, we construct position-independent and position–aware activity classifiers and compare their performance on our dataset (see Section 4.1.2).

The *position-independent* activity recognition approach simply consists of a single classifier per subject that is trained on all data independent of the device position. We expect this recognition approach to perform sub-optimal, as the motion information from the sensors can be assumed to be very different in the different positions for the same activity.

The *position-aware* activity recognition approach consists of a set of individual models for each device position and each subject. The model to be used is determined in a position recognition step that is executed before the actual activity recognition. Figure 4.9 provides an overview of the detection process: first, the unlabeled record is classified as a dynamic or a static activity. As mentioned above, this step is necessary as we can more reliably detect the device position if we know whether the current activity is a static or a dynamic activity. Then, the position of the device is recognized with an activity-level depended classifier that uses a feature set that has been optimized for the type of activity (i.e. dynamic or static). Finally, the current activity is recognized by selecting and applying the classifier for the detected device position. Obviously, the performance of the position-aware activity recognition approach relies on the correct identification of the device position. Therefore, to test the feasibility of this approach, we use the results of the activity-level dependent position detection experiments - including all mistakes made - as input for the activity recognition experiments.

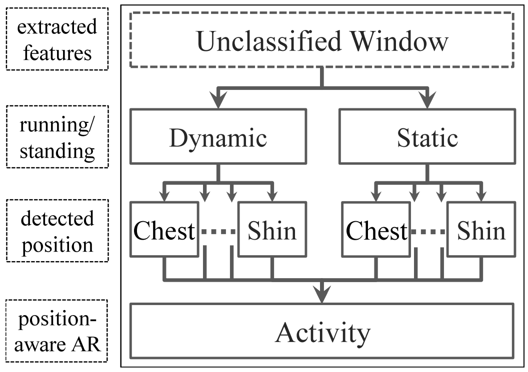


Figure 4.9: Physical Activity Recognition. The nodes illustrate the target class and the edges illustrate the applied models. The current window is classified as "dynamic" (climbing, jumping, running, walking) or "static" (standing, sitting, lying). Then the device position is recognized and a position specific classifier applied to derive the current activity.

4.3.3 Cross-Subjects Position-Aware Activity Recognition

The initial idea of a cross-subjects based model is to perform activity recognition also for people without corresponding training data or elderly which are unable to collect and label required data but, e.g., need to be observed. Commonly, a cross-subjects based approach relies on labeled sensor data of several people where the most known approach is leave-one-subject-out. Thus, a single classifier is trained on all available labeled data expect data of the target person. Compared to our single-subject approach, we focus on the performance of different cross-subjects approaches depending on the individual on-body device positions, i.e., in this scenario, we assume that we know the device position. However, we also evaluate how well the on-body positions are recognized in context of a cross-subjects based model. For that purpose and inspired by related works, we construct and evaluate the following cross-subjects approaches: *Randomly*, *Leave-One-Subject-Out*, *Top-Pairs*, and *Physical*. Especially, the physical-based approach could be promising as this idea was already hypothesized but not investigated in several previous works [108,122]. For all approaches, we follow a group-based approach where the groups are dynamically determined and can overlap for different subjects. Thus, a group represents certain people whose labeled data is considered to train a classification model for an unseen subject.

Leave-One-Subject-Out This approach was most often considered in related works (see Section 3.1.1) and performs often differently depending on the considered dataset. We build for each subject a classifier that relies on all available labeled data except the target person. We consider this approach as baseline.

Top-Pairs We compare our subjects pairwise to identify the best matches for each subject, i.e., we trained a classifier on data of one single person and evaluated the performance on another. Based on these results, we build a classifier for a target user that consists of the top five matches. In this context, it is unclear if the best matches taken together perform better or even worse due to contradictions. Indeed, this approach can only be evaluated if labeled data of the target person is available. For that purpose and in reference to our scenario, we consider only one minute per activity of the available labeled data of the target user.

Physical In initial experiments, we investigated whether demographic characteristics, in our case gender, fitness, and physique can be used to determine a group of people whose data can be used to recognize activities of a previous unseen subject. For this purpose, we identify these characteristics for each subject from our dataset. While gender and physique (strong and slim) were determined based on the videos of the exercises, we took the distance covered in 10 minutes running to cluster the subjects into five fitness levels. However, typically people do not have exactly the same physical characteristics but only some characteristics are similar. As a result, these people have comparable acceleration patterns for some activities but not for all. Hence, the choice of these characteristics based on the idea that people with the same fitness level have similar patterns concerning running while the gender and physique could be characterizing for walking. For clarification, Figure 4.10 illustrates the training and classification process and Figure 4.11 shows how we build the groups in respect of our dataset. For instance, if we want to build a classification model for subject 10 then we consider the labeled data of all subjects that are in the same row (same fitness level: 2, 4, 7, 13) or column (same gender and physique: 3, 9). In case that there is at most one match, we fallback and apply *leave-one-subject-out* (as this should be preferred compared to a pairwise approach [25]). In this context, we focused on a practical and feasible classification system to lower barriers and to enable an easy adoption.

Randomly As an additional reference, we also build classifiers where the number of considered people and also the people themselves are chosen at random except the target user. We repeat this approach ten times and consider the average as recognition rate.

During our experiments, we initially focus on dynamic activities because we believe that the acceleration patterns of static activities are less characterized by the individual behavior. We examine the performance and benefits of the introduced cross-subjects models but also the individual performance in context of each on-body position. Finally, we discuss and compare the results of our single-subject and cross-subjects approaches also in context of a multi-sensor setup.

4.3.4 Online Personalization of Cross-Subjects based Recognition Models

Online learning enables to evolve an existing model without keeping the whole dataset available. The model is adapted over time to the behavior of a user where recent received information is more weighted than older. In this context, we use online learning to adapt

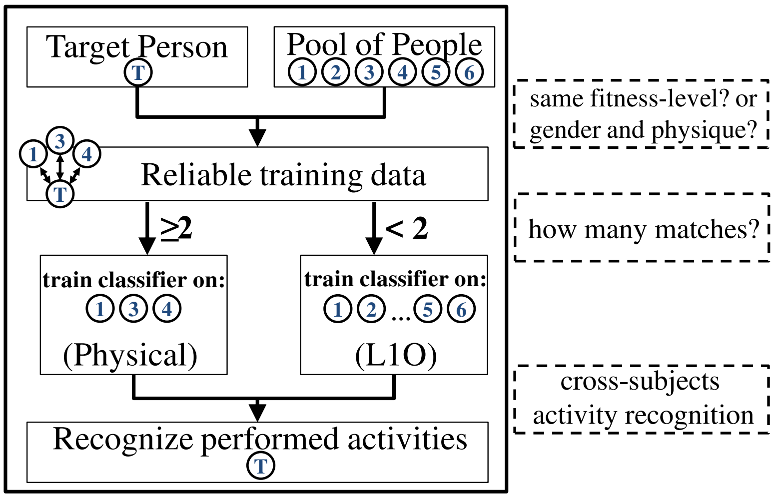


Figure 4.10: Cross-subjects activity recognition by relying on demographic characteristics, i.e., fitness, gender, and physique. For instance, to determine the activities of the target person T, we do this based on the known labeled data of subjects 1,3, and 4 (matches) which have the same fitness-level or gender and physique as T.

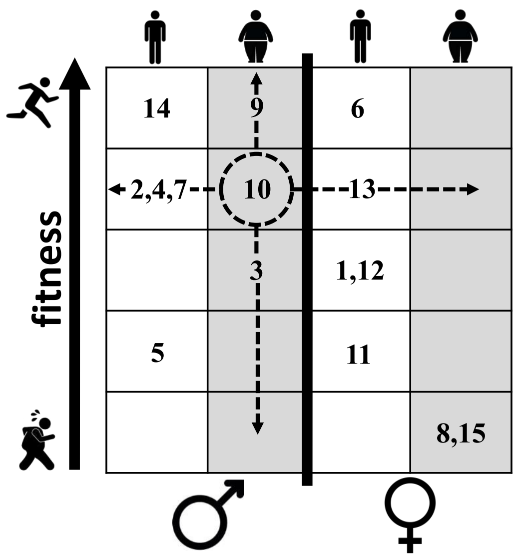


Figure 4.11: Cross-subjects activity recognition by relying on demographic characteristics. To identify suitable training data, we follow a group-based approach where the groups are dynamically determined and can overlap for different subjects. A subject has similar acceleration patterns to people in the same row and column.

a cross-subjects model by new information that is gathered from the classified windows. In the following, we introduce the techniques *smoothing* and *user-feedback* which we apply to gather this information. Both techniques are applied separately (see Figure 4.12).

We apply *smoothing* if a single classified window is surrounded by windows that belong to another activity. More precisely, if two preceding and two succeeding windows have the same class but another than the surrounded then the label is adjusted. The sample of the adjusted record is also used to update the model. Concerning *user-feedback* (active learning), we ask the user for feedback on certain samples that have been classified with a low confidence (as it usually has a high entropy). As it is unfeasible to ask the user for a specific window, we analyze and cluster the classified windows for a specific time interval. If several classified windows with a low uncertainty occur close to each other, we ask the user for that specific time interval. Based on preliminary experiments, we decided that a sequence of uncertain classified windows is interrupted if the distance between two uncertain windows is ≥ 5 seconds. Further, we only asked the user for feedback if a sequence was longer than 30 seconds. This value was chosen in respect of the amount of the available testing data. Figure 4.12 shows our approach in detail. The initial model classifies the acceleration data of the target user. Subsequently, the classified windows are analyzed to identify uncertain classified windows. These windows are used to gather new knowledge by *user-feedback* and *smoothing*.

The idea is that *smoothing* provides information regarding minor classification errors where *user-feedback* targets major classification errors. Hence, the resulting information from *user-feedback* and *smoothing* is combined to create a new, small, labeled dataset to update the initial model. To maximize the information gathering, we focused on classified windows with a low uncertainty. Of course, the number of uncertain windows depends on a predefined threshold. Hence, during our experiments, we also consider several different confidence value thresholds and analyze the relation between uncertainty, user interaction, and gained recognition rate.

To evaluate the improvement of our recognition model over time, we perform five iterations of this approach. In this context, an iteration comprises that first, the model has to process a certain amount of acceleration data where subsequently *user-feedback* and *smoothing* are performed separately. Afterwards, the model is updated with the gathered

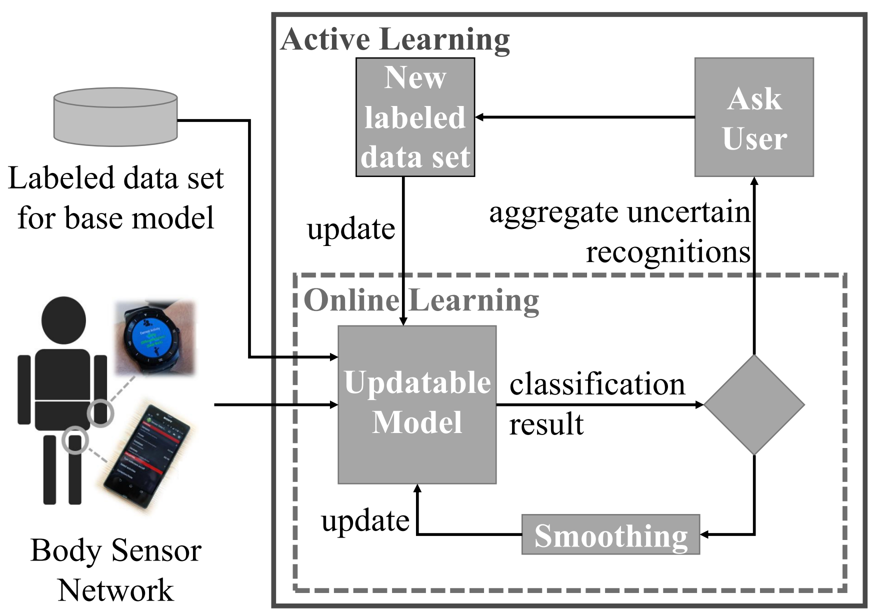


Figure 4.12: Personalization of a cross-subjects based model by online and active machine learning. This approach analyzes the classified windows regarding their uncertainty to gather new information.

data and the new performance is measured. To avoid overfitting, we separated the dataset of the target user in two equally sized parts where the classes are equally distributed. The one half is used to perform the introduced approach where the other half is considered to evaluate the performance of the evolving model. Hence, in each iteration, the model classifies new unseen acceleration data where the evolving model is always evaluated with the same dataset. We repeat our experiments several times where we also considered other splits of the datasets to make the results more stable. For these experiments, we rely on the introduced Online Random Forest classifier.

4.4 Experimental Results

In the following, we present our results and outline the conducted experiments to show the effect of the proposed methods. The presentation order is consistent compared to the introduced methods and the results are compared across the introduced approaches for discussion. Unless otherwise specified, the presented results are based on the Random Forest classifier which turned out to consistently perform better than other classification techniques. More detailed results are available as online resource[7]. Further, *F-measure* is considered as synonym of F_1-*measure*. In particular, we focus on the following research questions:

RQ1.1 Is it possible to recognize automatically the on-body position of a wearable device by the device itself?

RQ1.2 How does the information about the wearable device on-body position influence the physical activity recognition performance?

RQ1.3 Which technique can be used to build cross-subjects based activity recognition systems?

RQ1.4 Given a cross-subjects based activity recognition model, how can we adapt the model efficiently to the movement patterns of the user?

The following subsections belong to the publications [1, 3, 4].

4.4.1 Device On-body Localization

For the first experiment, we evaluated an activity-independent approach to create a baseline. Thus, we trained for each subject a single classifier on the data of all performed activities and each position. Table 4.3 shows the result and illustrates that the device position can be recognized with a F-measure of 81%. In this context, the *shin* (op_4) has the highest (88%) and the *forearm* (op_2) and *upper arm* (op_6) the lowest (79% / 78%) recognition rate. The latter highlights the problem regarding the flexibility of the arm

[7] `https://sensor.informatik.uni-mannheim.de/#results`

during each activity and also indicates that these two positions are the most problematic device locations. Examining the confusion matrix, shows that the individual positions are not mixed up. Indeed, the false-positives and the false-negatives are almost evenly distributed.

Table 4.3: Activity-independent position recognition rates for different on-body locations

Class	Precision	Recall	FP Rate	F-measure
op_1	0.79	0.82	0.04	0.80
op_2	0.79	0.78	0.03	0.79
op_3	0.79	0.82	0.04	0.80
op_4	0.90	0.86	0.02	0.88
op_5	0.83	0.80	0.03	0.82
op_6	0.79	0.78	0.03	0.78
op_7	0.79	0.81	0.04	0.80
avg.	0.81	0.81	0.03	0.81

Further investigations point to the fact that the recognition rate of the correct device location is higher if the related activity is characterized by stronger acceleration. Hence, the separation between static and dynamic activities results in a significantly different recognition rates for these two kinds of activity groups (static 72% /dynamic 89%). As we can see in Table 4.4, the recognition rate is consistently lower for static activities (−9%).

We examined the feature set and figured out that the gravity of the device provides useful information. However, attention should be paid to the fact that our experiments also showed that the gravity vector and derived features (roll and pitch) lead to overfitting. Hence, if a classifier was trained for a specific position then the position recognition rate dropped after the device was reattached for this position. This is mainly because the orientation of the device was slightly changed by the user. Thus, the orientation seems not to be a reliable indicator of the current device position. However, investigations have shown that static activities and the device orientation are correlated. Thus, the orientation enables to distinguish implicitly between the static activities, which results in less misclassifications of the device position across these activities. In this context, we only considered the introduced discretized orientation. Table 4.5 summarizes the results and shows that the recognition rate of the device localization in context of static activities increases by 16%.

Certainly, the usage of different feature sets for these two kinds of activity groups require the ability to separation between them. Hence, we constructed a classifier that

Table 4.4: Activity-level dependent position recognition rates showing that the recognition performance is problematic for static activities

Activities	Precision	Recall	FP Rate	F-measure
static	0.72	0.72	0.05	0.72
dynamic	0.89	0.89	0.02	0.89
both	0.81	0.81	0.03	0.81

Table 4.5: Position recognition rate for static activities and different feature sets showing that time-based and gravity-based features are needed to achieve an accurate recognition rate.

Features	Precision	Recall	FP Rate	F-measure
time-based	0.72	0.72	0.05	0.72
add'l gravity-based	0.88	0.88	0.02	0.88
only gravity-based	0.54	0.53	0.08	0.54

Table 4.6: Recognition rate for distinguishing between static and dynamic activities. The values represent the mean across all considered on-body positions.

Class	Precision	Recall	FP Rate	F-measure
dynamic	0.98	0.96	0.02	0.97
static	0.94	0.98	0.04	0.96
avg.	0.97	0.97	0.03	0.97

Table 4.7: Detailed results for the proposed on-body position recognition method. The values represent the mean across all considered physical activities.

Class	Precision	Recall	FP Rate	F-measure
op_1	0.87	0.89	0.11	0.88
op_2	0.87	0.85	0.15	0.86
op_3	0.86	0.89	0.11	0.87
op_4	0.95	0.92	0.08	0.94
op_5	0.91	0.90	0.10	0.91
op_6	0.85	0.84	0.16	0.85
op_7	0.91	0.92	0.08	0.92
avg.	0.89	0.89	0.11	0.89

decides to which activity group, the performed activity belongs. Table 4.6 outlines the result and clearly shows that the recognition performs very well (97%).

As a result, we evaluated the approach where we first decide if a static or dynamic activity is performed and then apply an activity-level specific position classifier. Compared to the baseline, Table 4.7 shows that this approach has an 8% higher recognition rate. In this context, the *shin* is still the best (94%) and the arm (*forearm* and *upper arm*) the worst (86% / 85%) position. Looking at the confusion matrix still exposes an evenly distribution of the false-negatives and false-positives but certainly lower values. This indicates that the distinction of the activity-levels, more precise, the individual handling of the dimensions of the data lead to a better distinction of the device positions. Hence, the experiments shows that in most of the cases it is possible to recognize the device position correctly. Thus, in general the considered positions seem not to be mixed up concerning the classification which confirms that each position provides different information for the same activity.

In summary, our on-body position recognition approach that makes use of a Random Forest classifier and distinguishes between different activity levels achieves an average performance of 89% across all positions.

4.4.2 Single-Subject Position-Aware Activity Recognition

The whole concept is based on the idea that knowledge about the device position improves activity recognition. Therefore, we also have to show that the position-aware activity recognition approach that uses the automatically detected device position outperforms the baseline approach that does not consider the device position. For this purpose, we constructed and examined the introduced position-independent activity classifier for each subject which was trained on all data of all positions. Table 4.8 illustrates the performance of this approach and shows that the correct activity is recognized with a F-measure of 80%. However, considering the individual activities, it shows that the recognition rate is unequally distributed. Thus, *sitting* (ac_6) has a significantly worse (67%) and *jumping* (ac_3) a much better (96%) recognition rate. Additionally, the activities *climbing down* (ac_1) and *standing* (ac_5) are often confused with other activities. In this context, the related confusion matrix (see Figure 4.13) emphasizes that the recognized activity is often wrong if a performed activity is similar to another, i.e., *lying* (ac_4), *standing* (ac_5), and

Table 4.8: Results of the baseline method for activity recognition without position information.

Class	Precision	Recall	FP Rate	F-measure
ac_1	0.84	0.76	0.02	0.80
ac_2	0.77	0.81	0.04	0.79
ac_3	0.99	0.94	0.00	0.96
ac_4	0.83	0.83	0.03	0.83
ac_5	0.68	0.77	0.06	0.72
ac_6	0.70	0.64	0.05	0.67
ac_7	0.93	0.89	0.01	0.91
ac_8	0.85	0.87	0.03	0.86
avg.	0.80	0.80	0.03	0.80

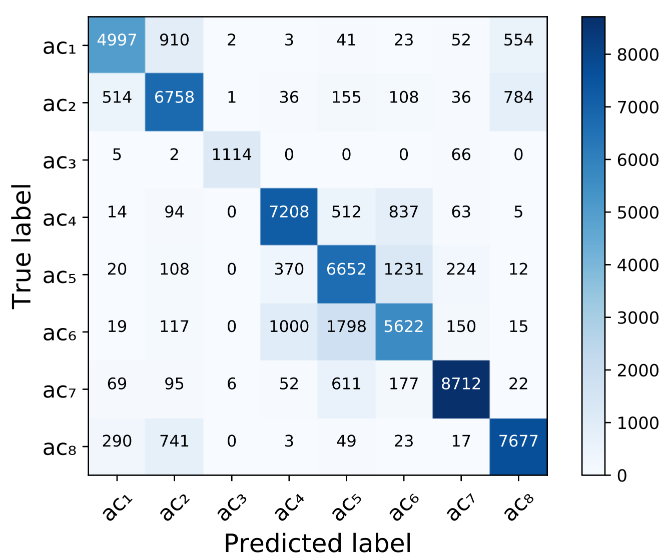


Figure 4.13: Confusion matrix for the baseline activity recognition method without position information.

Table 4.9: Results of the proposed activity recognition method that uses automatically detected device positions.

Class	Precision	Recall	FP Rate	F-measure
ac_1	0.84	0.77	0.02	0.81
ac_2	0.78	0.81	0.04	0.79
ac_3	0.99	0.95	0.00	0.97
ac_4	0.90	0.88	0.02	0.89
ac_5	0.74	0.81	0.05	0.77
ac_6	0.78	0.74	0.04	0.76
ac_7	0.94	0.91	0.01	0.92
ac_8	0.85	0.88	0.03	0.86
avg.	0.84	0.83	0.03	0.84

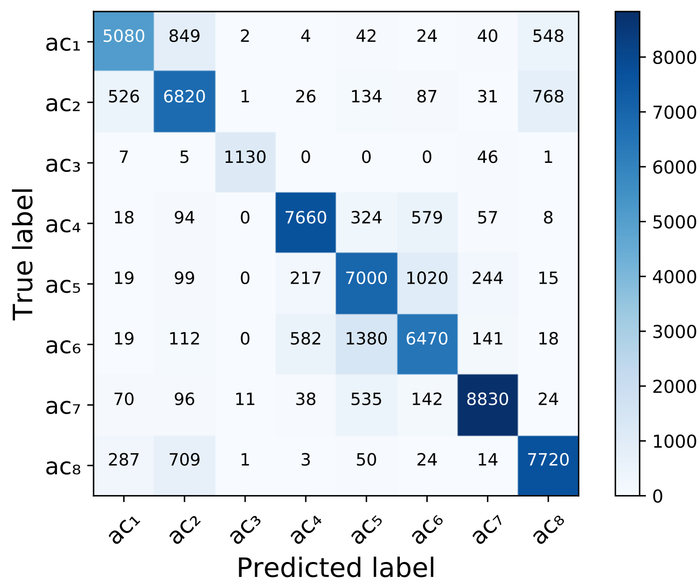


Figure 4.14: Confusion matrix: Proposed activity recognition method using automatically detected device position.

sitting (ac_6) but also *climbing up* (ac_1), *down* (ac_2), and *walking* (ac_8) are often mixed up.

In contrast, the introduced position-aware approach achieves a 4% higher F-measure. Table 4.9 shows that for each activity, the consideration of the on-body device position results in a higher or equal recognition rate. Concerning the static activities, we can observe that the F-measure values increased significantly. Indeed, the activities *lying* (+6%), *standing* (+5%), and *sitting* (+9%) have improved the most. In this context, the related confusion matrix (see Figure 4.14) makes clear that the problem of misclassification is not completely solved but better handled than before. For dynamic activities, the recognition rate improved slightly.

Considering the activities and positions in detail (see Table 4.10), it leads to the fact that there is no optimal device position. The chest, waist, thigh, and shin perform on average at best but they perform different depending on the activity. Thus, the

Table 4.10: Results (F-measure) of the proposed activity recognition method with known device positions.

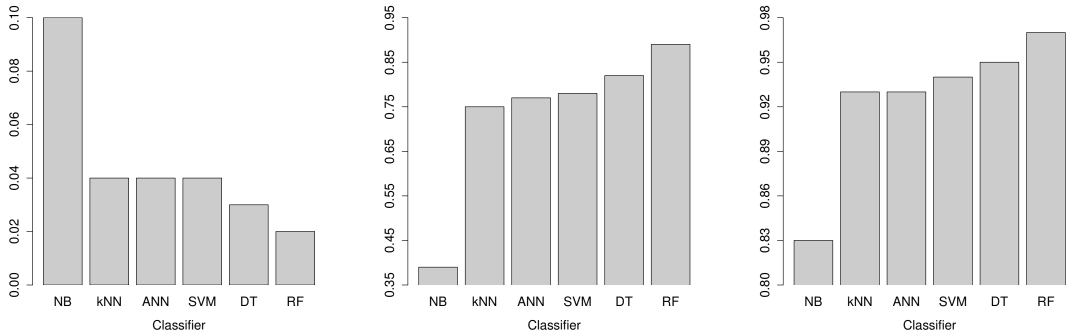
Class	op_1	op_2	op_3	op_4	op_5	op_6	op_7
ac_1	0.86	0.75	0.76	0.83	0.81	0.80	0.82
ac_2	0.83	0.72	0.76	0.84	0.83	0.78	0.80
ac_3	0.97	0.97	0.97	0.95	0.95	0.98	0.97
ac_4	0.89	0.83	0.89	0.90	0.86	0.94	0.91
ac_5	0.72	0.73	0.71	0.86	0.84	0.75	0.81
ac_6	0.72	0.76	0.65	0.82	0.80	0.74	0.82
ac_7	0.92	0.91	0.91	0.93	0.94	0.92	0.93
ac_8	0.89	0.82	0.82	0.89	0.88	0.85	0.88
avg.	0.84	0.80	0.79	0.87	0.86	0.83	0.86

activity *climbing stairs up* is best handled by the chest (up to 5% better) whereas the *thigh* recognizes the activity *standing* the best (up to 14% better). This confirms a statement of related work where they stated that the optimal sensor placement depends on the activity [34]. Further, it points out that most of the positions perform still bad regarding the static activities. This indicates that even low acceleration combined with the (predicted) device position makes it hard to distinguish between such activities. Besides, there are also activities where each position performs very well. Hence, the activities *running* ($\geq$ 91%) and *jumping* ($\geq$ 95%) are equally well recognized for all positions due to the high acceleration of the devices. These show that the acceleration strength is decisive concerning the activity recognition rate and that in case of low acceleration additional information of the environment or context-related information are required.

Despite the fact that we recognized only in 89% of all cases a correct device position and compared with the position-independent approach (80%), these results indicate clearly that the consideration of the device position results in a higher activity recognition rate (84%). The results show that it does not depend on the activity but on the device position if the information of the device position improves the activity recognition rate. In this context, also the individual handling of the different dimensions (e.g., device position and static/dynamic activities) leads to a better distinction of the target classes, so to a better recognition rate. Especially in context of static activities, these two approaches lead to a significant better recognition.

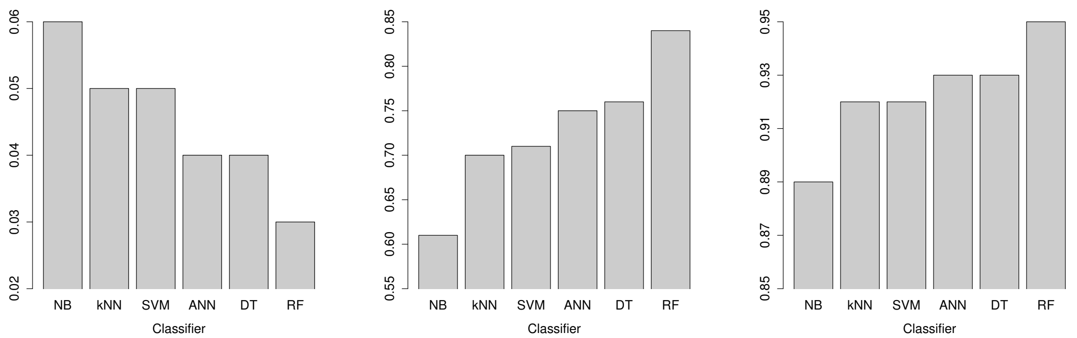
In order to show the benefits of using the proposed Random Forest classifier, we compared its performance with other common classification methods, in particular Artificial Neural Network (ANN), Decision Tree (DT), k-Nearest Neighbors (kNN), Naive Bayes (NB), and Support Vector Machine (SVM). All of these classifiers were used in previous work on activity recognition and they achieved good results.

Considering the activity-level (static/dynamic) depended on-body position recognition approach, the other classifier performed worse. Figure 4.15 illustrates the results and shows clearly that *Random Forest* (89%) outperforms the other classifier. In this context, *NB* (39%) performed the worst probably due to assumption that all features are independent. In contrast, *k-NN* (75%), *ANN* (77%), and *SVM* (78%) achieved reason-



(a) FP-Rate **(b)** F-measure **(c)** Accuracy

Figure 4.15: Performance of the different classifier for position recognition in the activity-level (static/dynamic) dependent scenario.



(a) FP-Rate **(b)** F-measure **(c)** Accuracy

Figure 4.16: Performance of the different classifier for position-aware activity recognition. The on-body device position was detected in a previous step by the activity-level (static/dynamic) dependent approach (using Random Forest).

able results. We performed parameter optimization and choose a radial basis function regarding *SVM*. The *DT* (82%) performed second best but the recognition rate is much worse (−7%) than that of the *RF*. Besides, the training phase of the *RF* was one of the fastest whereas *ANN* and *SVM* took the longest.

Concerning activity recognition, we evaluated the performance of the classifier in context of position-aware activity recognition based on the recognized device positions of the Random Forest. Figure 4.16 shows that *RF* (84%) achieved the highest activity recognition rate where *NB* (61%) performed the worst. Further *k-NN* (70%) and *SVM* (71%) performed almost equal but worse than *ANN* (75%) and *DT* (76%). Besides, we also evaluated the performance of all classifier in a position-independent scenario but it expose that independent of the classifier the position-aware approach is always better.

These results show that the use of the Random Forest classifier is not only the best classification method for determining the device position, it also outperforms all other classifiers with respect to determining the activity given a hypothesis about the position of the device.

4.4.3 Cross-Subjects Position-Aware Activity Recognition

In several cases, people are unable to collect and label data which is required for a subject-specific approach. Therefore, we also focused on the feasibility to recognize the performed activity and device position by relying only on labeled sensor data of other people. For that purpose, we evaluate the performance of the introduced cross-subjects approaches *randomly*, *leave-one-subject-out* (L1O), *top-pairs*, and *physical*. We aim to clarify how differently these approaches perform but also the performance in general depending on the device position and compared to a subject-specific approach. In this context, preliminary experiments already clarified that cross-subjects based recognition models perform worse than single-subject based models. For that reason, we also investigate setups with multiple accelerometers to determine if it is possible to reach a comparable recognition rate by using more acceleration sensors. Unless otherwise specified, the provided results are based on the Random Forest classifier which turned out to consistently perform better than other classification techniques (cf. see Section 4.4.2).

4.4.3.1 Activity Recognition with a Single Accelerometer

During the first experiments, we only consider dynamic activities as target classes to avoid misinterpretation. Thus, we assume that static activities are less characterized by an individual person, i.e., the subtle acceleration that is performed by these activities is probably similar for many different groups of people.

Table 4.11: Dynamic activity recognition (F-measure): Performance of cross-subjects approaches on each individual device position. Each classifier was only trained and tested with data of a specific on-body position (single accelerometer).

Position	Randomly	L1O	Top-Pairs	Physical
op_1	0.64	0.70	0.69	0.68
op_2	0.60	0.66	0.64	0.65
op_3	0.56	0.62	0.61	0.61
op_4	0.63	0.70	0.71	0.70
op_5	0.54	0.58	0.58	0.59
op_6	0.65	0.72	0.71	0.72
op_7	**0.69**	**0.76**	**0.77**	**0.78**

As a first step, we focused on the activity recognition rate of position-dependent classifiers to expose differences in performance. Table 4.11 shows that across all positions, the introduced approaches perform comparable but the recognition rate varies significantly. The waist seems to be the best on-body position for all approaches where *physical* achieves the highest activity recognition rate (78%). In this context, the results indicate that the acceleration patterns for the same activity across several users are most similar at this position. Considering the baseline (*L1O*), *top-pairs* (+1%) and *physical* (+2%) perform slightly better while they have to process significantly less data. Besides, previous work already showed that *L1O* would not scale in a large-user environment due to the varying behavior. Actually, the classifier seems only to learn the dominant behavior across all people, i.e., individual behavior is lost and rated as noise. Considering the other positions, it points out that surprisingly the thigh (op_5) based classifier performs the worst. We examined the individual acceleration patterns and detected that the bad performance results from the unstable position of the device (trouser pocket). Hence, the device was able to move slightly during the data collection. This kind of noise could be handled by a subject-specific approach because it was consistent but this is not the case across subjects. However, this does not mean that the position is unsuitable but, e.g., needs more effort concerning personalization (cf. [135]).

Table 4.12: Dynamic activity recognition rate (F-measure) for each cross-subjects approach: The classifiers were only trained on data that belongs to the waist (op_7).

Class	Randomly	L1O	Top-Pairs	Physical
ac_1	0.62	0.65	**0.69**	**0.69**
ac_2	0.62	**0.70**	**0.70**	**0.70**
ac_3	0.75	**0.83**	0.82	0.78
ac_7	0.87	0.89	**0.92**	0.91
ac_8	0.63	0.76	0.75	**0.78**
avg.	0.69	0.76	0.77	**0.78**

Considering the recognition rate of the individual activities, Table 4.12 shows the corresponding recognition rates of the waist-based classifier. Independent of the evaluated approaches, *climbing stairs* (∼70%) has the lowest and *running* (∼91%) the best recognition rate. Indeed, compared to *L1O*, it points out that all activities expect jumping are best recognized by *physical*. In this context, especially *climbing stairs* and *walking* have a higher recognition rate. This is remarkable because these are the only dynamic activities which are most often confused. We believe that this is evidence for the feasibility to rely on common physical characteristics to identify meaningful groups. However, we also conclude that our considered physical characteristics do not cover the features of *jumping*. Besides, *top-pairs* performs slightly better than *L1O* but, e.g., concerning *walking* even worse. We noticed during the experiments that the acceleration patterns were contradictory while the classifier learned the dominant behavior.

Finally, we also considered static activities (ac_4-ac_6). Table 4.13 shows that the recognition rate seems to be stable but the recognition rate of dynamic activities drops slightly. During this experiment, we also applied the introduced static and dynamic activity split (including all errors) to consider the gravity based feature in context of static activities. On the one hand, this division caused a decrease of the dynamic activity recognition rate, on the other hand the confusion matrix shows (not presented) that especially lying (ac_4) and standing (ac_5) are significantly less confused due to the considered gravity based features. Thus, the results indicate that these features are also reliable across people. Compared to our single-subject approach (see Table 4.10), especially the recognition of *climbing stairs* performs worse whereas the recognition rate of static activities is comparable (±2%). This confirms our initial assumption concerning static activities in context of cross-subjects models.

Table 4.13: Static and dynamic activity recognition rate (F-measure) using the *physical* approach (only waist (op_7), best performing position).

Class	Precision	Recall	F-measure
$\boldsymbol{ac_1}$	0.70	0.67	0.68
$\boldsymbol{ac_2}$	0.71	0.69	0.70
$\boldsymbol{ac_3}$	0.73	0.84	0.78
$\boldsymbol{ac_4}$	0.98	0.92	0.95
$\boldsymbol{ac_5}$	0.69	0.82	0.75
$\boldsymbol{ac_6}$	0.76	0.80	0.78
$\boldsymbol{ac_7}$	0.91	0.78	0.84
$\boldsymbol{ac_8}$	0.77	0.79	0.78
avg.	0.79	0.79	0.79

4.4.3.2 Activity Recognition with Two Accelerometers

To address the difference in performance, we also analyzed the improvement that can be achieved by an additional acceleration sensor. After all, several people already wear two devices. In the following, we exclude *Top-Pairs* as it is not feasible (see Section 4.3.3) and the preceding results indicate nothing remarkable.

Table 4.14 illustrates the possible improvement if we combine two of the best performing on-body device positions (shin and waist). In average, the recognition rate increases by 3% where especially the recognition of climbing stairs improved (+5%). On the downside, *walking* only increased slightly. However, this also makes clear that this activity is challenging. In this context, Figure 4.17 shows the corresponding confusion matrix. It strikes that the problematic groups are still *climbing up* (ac_1), *climbing down* (ac_2), *walking* (ac_8) and *lying* (ac_4), *sitting* (ac_5), *standing* (ac_6). Compared to our single-subject approach, it points out that no new issues arise but existing will become more manifest, e.g., jumping is more often confused with running.

Subsequently, we also investigated the recognition rate for different combinations of sensors that are realistic in a real world setting, in particular thigh and forearm (smart-phone and smart-watch) and thigh and head (smart-phone and smart-glasses). Table 4.15 summarizes these results. As we can see, these interesting combinations (smart-phone and

Table 4.14: Improvement of the activity recognition rate (physical approach) with an additional accelerometer (shin (op_4) and waist (op_7), cf. see Table 4.13).

Class	Precision	Recall	F-measure
$\boldsymbol{ac_1}$	0.72	0.74	0.73
$\boldsymbol{ac_2}$	0.72	0.75	0.74
$\boldsymbol{ac_3}$	0.83	0.92	0.87
$\boldsymbol{ac_4}$	0.99	0.92	0.95
$\boldsymbol{ac_5}$	0.74	0.88	0.80
$\boldsymbol{ac_6}$	0.80	0.86	0.83
$\boldsymbol{ac_7}$	0.94	0.79	0.86
$\boldsymbol{ac_8}$	0.83	0.75	0.79
avg.	0.83	0.81	0.82

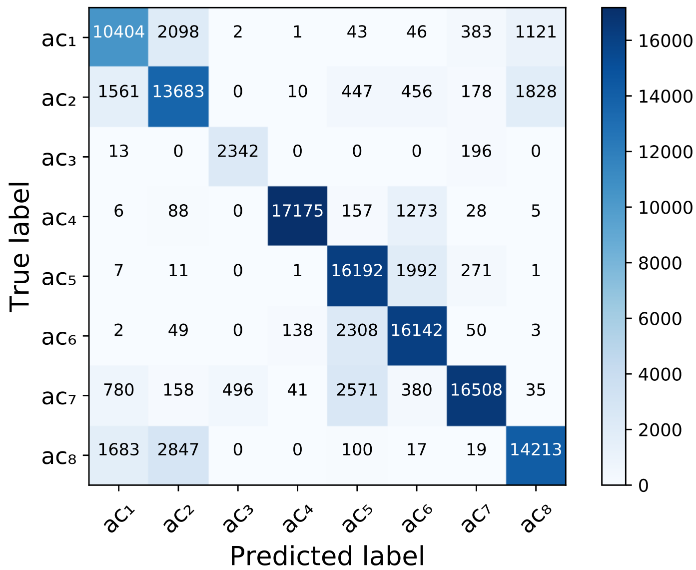


Figure 4.17: Confusion matrix: Two accelerometers (shin (op_4) and waist (op_7)), cross-subjects based approach (physical), cf. see Table 4.14.

smart-watch (69%) and smart-phone and smart-glasses (72%)) perform significantly worse than the best two-sensor combination (see Tables 4.14 and 4.15). This indicates that a cross-subjects based model needs personalization to be applicable in a real-world setting. In this context, it also points out that it depends on the set of activities that should be recognized which combination is most suitable. Further, as we analyzed the individual activities concerning all on-body device positions and combinations and in each case, the physical approach performs equal or better, we can state these results provide evidence that the considered physical characteristics are reliable properties to identify which people can be considered for a group-based cross-subjects model. Certainly, due to the size of our dataset, it is likely that there are further meaningful characteristics which we could not identify. However, these results confirm the hypothesis of previous works [108, 122].

For completeness, Table 4.16 shows the average recognition rates of all possible two-part accelerometer setups of the different approaches. We can see that the *physical* ap-

Table 4.15: Recognition rates of interesting accelerometer/position combinations (our approach).

	op_2-op_5 (Watch & Phone)			**op_3-op_5 (Glasses & Phone)**		
Class	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
ac_1	0.61	0.58	0.59	0.44	0.61	0.51
ac_2	0.56	0.74	0.64	0.65	0.72	0.69
ac_3	0.99	0.87	0.93	0.99	0.75	0.85
ac_4	0.64	0.39	0.48	0.83	0.77	0.80
ac_5	0.84	0.80	0.82	0.77	0.79	0.78
ac_6	0.48	0.70	0.57	0.64	0.67	0.66
ac_7	0.98	0.97	0.98	0.96	0.93	0.94
ac_8	0.77	0.61	0.68	0.74	0.48	0.58
avg.	0.71	0.69	0.69	0.74	0.72	0.72

Table 4.16: Results (F-measure) show the recognition rates for the individual activities of the cross-subjects approaches (average of all possible two-setup combinations).

Class	Randomly	Leave-one-out	Physical
ac_1	0.62	0.66	**0.69**
ac_2	0.63	0.67	**0.69**
ac_3	0.79	**0.88**	0.87
ac_4	0.81	0.83	**0.86**
ac_5	0.71	0.73	**0.79**
ac_6	0.59	0.63	**0.68**
ac_7	0.88	0.90	**0.96**
ac_8	0.60	0.67	**0.70**
avg.	0.69	0.74	**0.78**

proach performs overall satisfying in respect of all activities. Focusing on static (77.7%) and dynamic (78.2%) activities separately, points out that their recognition rates are similar but the rates for climbing stairs (ac_1 and ac_2, 69%) and walking (ac_8, 70%) are lower. Varying movement speed and patterns of these activities cause these lower recognition rates. In contrast, running (ac_7) and jumping (ac_3) have significantly higher recognition rates because the strong acceleration is a reliable indicator. Indeed, considering the confusion matrix (not presented), climbing stairs and walking are activities that are often confused among each other. This problem seems to occur independently of the number of accelerometers.

4.4.3.3 Activity Recognition with Multiple Accelerometers

Finally, we examine the relation between number of accelerometers and the activity recognition rate. Table 4.17 shows the results of the corresponding experiments and indicates that our physical-based approach still consistently performs better than the other approaches (+3%) where *randomly* produces the worst results (−4.5%). Indeed, with an increasing number of accelerometers the gap between the recognition rates seems to remain stable. The results also show that the recognition rates are far worse than a subject-specific classifier (see Table 4.9). At least a four-sensor setup seems to be necessary to achieve even satisfying recognition rates. This is not feasible in a real world scenario and underlines the necessity for adapting the model to new individuals.

If we shift our focus to a scenario where we could rely on additional wearable devices, Figure 4.18 shows the improvements concerning the different activities. Indeed, considering all activities, a two-part setup performs always better than a single sensor independent

Table 4.17: Recognition rates (F-measure) of the introduced cross-subjects based approaches.

	Number of Accelerometers					
	1	2	3	4	5	6
Randomly	0.61	0.69	0.75	0.77	0.79	0.80
Leave-one-out	0.65	0.74	0.79	0.82	0.83	0.85
Physical	**0.68**	**0.78**	**0.82**	**0.85**	**0.87**	**0.88**

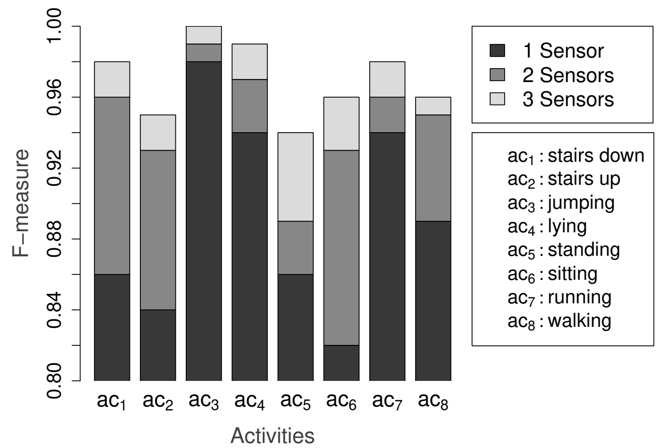


Figure 4.18: The recognition rates of a multi-sensor setup. It illustrates the possible improvements of the recognition rate for each activity.

of the chosen on-body device positions. Hence, the worst two-part setup (head and upper arm) still achieves a recognition rate of $\geq 90\%$ where the best combination (tight and waist) has up to 94%. Besides, the worst combinations always cover a position which is located on the arm or on the head. This is consistent with the preceding results, i.e., it is due to the flexibility. In contrast, the best two-part combinations consist always of the sensors which performed the best in a single sensor environment. All of this also holds if we compare a three- and two-part setup.

Considering the individual physical activities, the biggest improvements with a two-part setup could be achieved concerning *sitting* ($ac_6, +11\%$), *climbing stairs* ($ac_1, +10\%$ and $ac_2, +9\%$) and *walking* ($ac_8, +6\%$). This is strong evidence that already one additional wearable device increases the robustness and quality of the recognition system significantly. Further, it does not matter if the on-body position selection is up to the subject. A third sensor still improves the recognition for all activities but less significant.

4.4.3.4 Device On-body Localization

Finally, we also investigated if cross-subjects based models are able to recognize the on-body device position. Table 4.18 shows the individual recognition rate. Independent of the approach, it points out that the recognition quality differs significantly across the different positions where waist (78%) and shin (74%) are best recognized. Considering the overall results, we have to state that the position recognition rates are not sufficient to be considered as part of an activity recognition system. However, these results also confirm our assumption that the waist seems to be the best on-body device position for cross-subjects activity recognition.

In general, the results show that cross-subjects models are feasible for activity recognition if the on-body device position is known a-priori. In this context, the waist is the best device position for cross-subjects activity recognition where we were able to achieve

Table 4.18: Activity-independent position recognition (F-measure): Performance of cross-subjects approaches concerning the recognition of the on-body device position (single accelerometer).

Class	Randomly	L1O	Top-Pairs	Physical
op_1	0.56	0.63	0.59	0.61
op_2	0.58	0.63	0.59	0.58
op_3	0.54	0.61	0.56	0.57
op_4	0.68	0.74	0.72	0.73
op_5	0.53	0.60	0.57	0.58
op_6	0.50	0.57	0.53	0.54
op_7	0.74	0.78	0.76	0.77

a recognition rate of 79%. Considering an additional wearable device, improved the performance by +3%. Thus, our results indicate that it is feasible to monitor the physical activities of people which are unable to collect and label required data. Further, the *physical* based approach performed the best in context of the most reliable device position where especially walking and climbing stairs are better handled. Besides, we consider the recognition of the device position in a cross-subjects scenario still as open issue which needs further investigations.

4.4.4 Online Personalization of Cross-Subjects based Recognition Models

To modifying existing classification models without re-training, i.e., to adapt the model to the user's behavior, the classifier has to operate in online instead of offline mode. For that reason, in the following we rely on the introduced *Online Random Forest* (see Section 2.3.4.6), as such, we also investigate the gap in performance between the online and offline mode of the Random Forest.

The core idea is that feedback concerning the classification results improves the cross-subjects based activity recognition model. To confirm this theory, we performed a series of experiments in improving the physical-based recognition models using online and active learning. More precisely, first we analyze the difference in performance regarding offline and online learning. Subsequently, we investigate our introduced information gathering methods, i.e., *user-feedback* and *smoothing*, to personalize the model. Finally, we focus on the obtained activity recognition rate concerning certain aspects.

Table 4.16 and 4.19 illustrate the activity recognition rate for our approach in offline and online mode. It points out that in online mode the recognition rate is slightly worse (−2%). This is due to fact that in online mode the classifier does not know the whole dataset a priori. Therefore, the chosen internal thresholds of the classifier concerning the node splits and features are coarser. In turn, this ensures that the trained classifier is not fitted to a specific dataset. Further, the lower initial recognition rate of the base model is the drawback to enable to update the model on the fly without knowing or storing preceding data.

Table 4.19: Online and active learning: Improvements of the recognition rate (F-measure) concerning personalization of the base model.

Class	Our method (Base)	+ Smoothing	+ User-Feedback	+ Smoothing & User-Feedback
ac_1	0.65	0.67	0.80	0.80
ac_2	0.66	0.68	0.80	0.81
ac_3	0.82	0.87	0.89	0.90
ac_4	0.86	0.86	0.88	0.88
ac_5	0.77	0.77	0.79	0.79
ac_6	0.66	0.66	0.70	0.70
ac_7	0.95	0.96	0.97	0.97
ac_8	0.71	0.74	0.86	0.87
avg.	0.76	0.78	0.83	0.84

Applying our personalization approach (*smoothing* & *user-feedback*) improves the recognition rate of the base model by +8% (see Table 4.19). Considering the individual activities show that the recognition rate improves for all activities (up to +16%). If we examine static and dynamic activities separately (see Table 4.20), it strikes that the recognition rate improves especially for dynamic activities (+11%) where the performance concerning static activities increases slightly (+3%). This means that the dynamic activities are much better characterized by acceleration data and that even the gravity-based features that we took into account for static activities did not resolve this issue. The corresponding confusion matrix (see Figure 4.19) confirms this statement. Hence, the static activities lying (ac_4), standing (ac_5), and sitting (ac_6) are often confused among each other. Even *user-feedback* only improves the recognition of these activities slightly. In contrast, the dynamic activities also cover activities that are confused (climbing down (ac_1), climbing up (ac_2), and walking (ac_8)) but the *user-feedback* mostly resolves this problem.

Evaluating these two techniques separately and together showed that they improve different parts of the activity recognition model thus complementing each other (see Table 4.19 and 4.20). Focusing only on *smoothing*, the performance of the base model improves by ∼1-2% where mostly the recognition rate of dynamic activities increased. This indicates that this kind of minor errors occur less frequency. Indeed, the more acceleration data was processed by our updatable model, the less frequently such errors occurred.

Table 4.20: Distinction between static and dynamic activities concerning online and offline training.

	Static			Dynamic		
Method	Precision	Recall	F-measure	Precision	Recall	F-measure
Our approach (offline)	0.78	0.77	0.78	0.79	0.78	0.78
Our approach (online)	0.77	0.76	0.76	0.76	0.75	0.76
+ Smoothing	0.79	0.79	0.79	0.88	0.85	0.86
+ User-Feedback	0.80	0.79	0.79	0.86	0.86	0.86
+ Smoothing & U-F	0.80	0.79	0.79	0.88	0.86	0.87

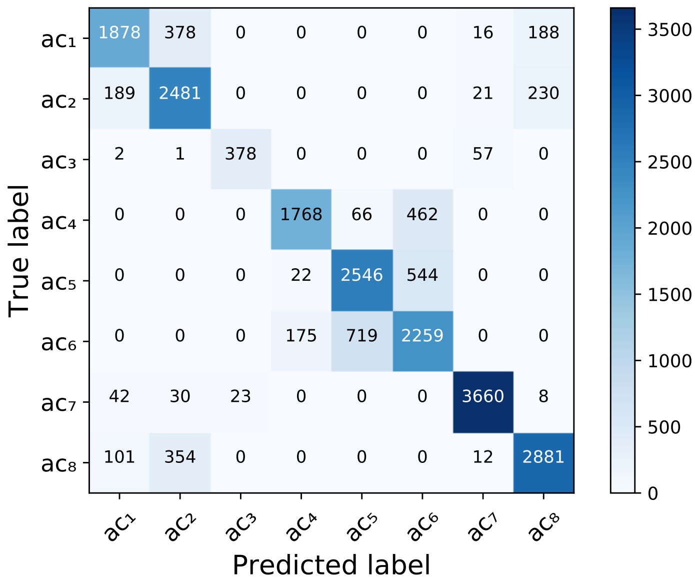


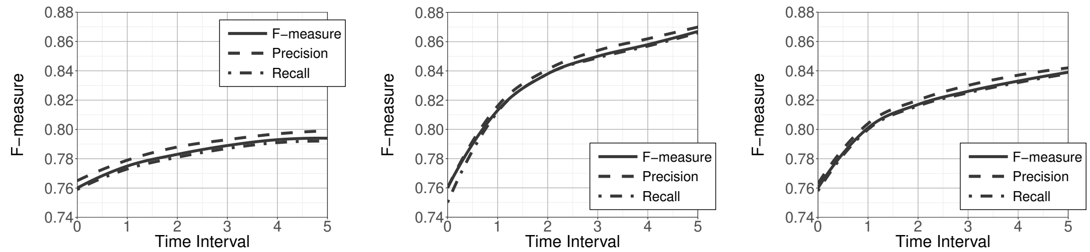
Figure 4.19: Confusion matrix after the personalization (smoothing and user-feedback) of the base model (our approach, cross-subjects, two accelerometers) with online and active learning. The presented values are divided by 100 and rounded.

Table 4.21: Online and active learning: After personalization of the base model (our approach): Recognition rates of interesting accelerometer/position combinations.

	op_2-op_5 (Watch & Phone)			op_3-op_5 (Glasses & Phone)		
Class	**Precision**	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
ac_1	0.80	0.72	0.76	0.79	0.77	0.78
ac_2	0.77	0.81	0.79	0.82	0.84	0.83
ac_3	0.98	0.87	0.92	0.97	0.83	0.89
ac_4	0.83	0.61	0.70	0.90	0.79	0.84
ac_5	0.82	0.82	0.82	0.77	0.89	0.83
ac_6	0.59	0.75	0.66	0.73	0.72	0.73
ac_7	0.98	0.98	0.98	0.97	0.97	0.97
ac_8	0.83	0.86	0.85	0.87	0.88	0.87
avg.	0.81	0.80	0.80	0.84	0.84	0.84

Focusing on the same specific device position combinations as in the previous section (see Table 4.15 and 4.21), it points out that also for these combinations the recognition rate improved significantly (watch & phone (+11%), glasses & phone (+12%)). Considering the individual activities, especially walking (ac_8) achieves a satisfying recognition rate (85% and 86%). As in the preceding results, jumping (ac_3) and running (ac_7) have the highest and sitting (ac_6) the lowest recognition rates.

The personalization of a cross-subjects model is a continuous process. Figure 4.20 shows how the performance evolves over time and clarifies that especially the recognition rate of dynamic activities improves significantly (87%). Each time interval covers acceleration data for each activity and also the same amount of data across the intervals that are classified by our model. For both activity types, we can observe that the recognition rate increased mostly during the first two time intervals. This indicates that the number of



(a) Static activities **(b)** Dynamic activities **(c)** All activities

Figure 4.20: Static vs. dynamic activity recognition: Improvement due to active learning of the base recognition model (our approach) over time.

windows with a low confidence classification decreases with each iteration. The recognition rate of static activities seems to converge which is an indicator that the acceleration data is not sufficient. Nevertheless, the recognition rate of the base model improves after the first iteration by +4% and after five iterations by +8% (84%).

We also evaluated different thresholds for the confidence value of the classified windows. Figure 4.21 shows the ratio between additional obtained recognition rate (first derivative, slope) and the number of questions that has to be answered by the target person. It depicts that a higher confidence value results in a larger number of classified windows that are considered as uncertain so the number of questions increases. Of course, the number of questions depends on the number of considered activities, more precisely, the number of activity instances that are covered by the considered dataset. During our experiments, we assumed that all considered activities occurred exactly once during a *time interval*. For our presented results, we considered a threshold of 0.5 to keep the number of questions small but cover the turning point of the slope. Hence, in average each user had to answer ∼10 questions to improve the base recognition model by +8%. Besides, if the threshold is high, the slope function converges to zero, i.e., windows with a high confidence value are correct classified.

Finally, we examined the relation between the activity recognition rate and the number of trees of an Online Random Forest classifier (see Figure 4.22). It points out that already a forest with 10 trees performs comparable to a forest with 100 trees. Indeed, their recognition rate differs only by ∼1-2% where precision and recall are close to each other. The advantages which result from a small forest are less computational power, lower memory usage, and a shorter computation time. This result shows the feasibility of online learning on wearable devices.

All of these results are a strong evidence for the feasibility that cross-subjects based models can be personalized by online and active machine learning. The personalized models achieve recognition rates of 84% and for dynamic activities even 87%. Concerning static activities, gravity-based features enable to decrease the confusion between standing and lying where sitting is still often confused with these two activities. Further, instead of collection a labeled dataset, the personalization of an existing base model is significantly less effort for the target user and also feasible for elderly and patients. Besides, the achieved recognition rates are comparable to subject-specific approaches of previous works [1, 35].

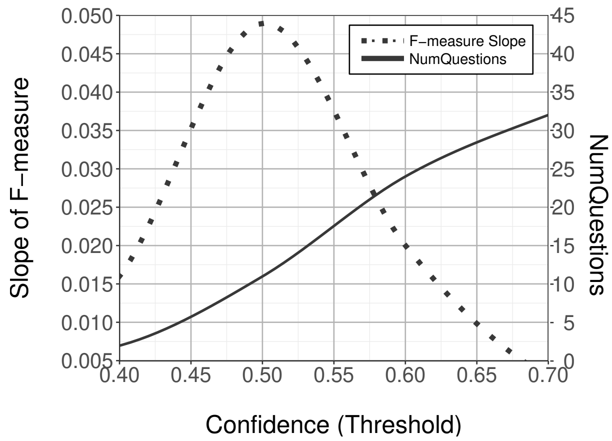


Figure 4.21: Progression of the activity recognition rate dependent on the confidence threshold concerning uncertain windows.

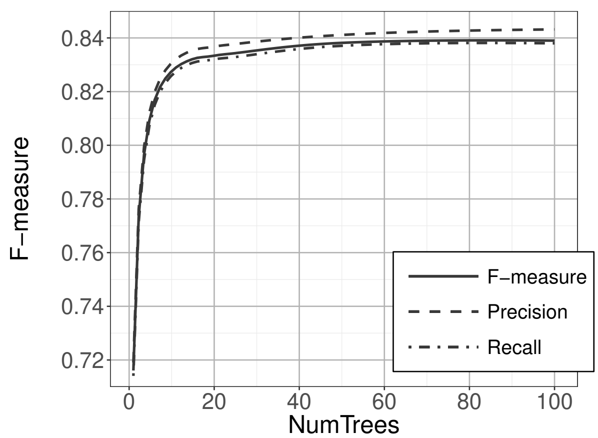


Figure 4.22: Influence of the size of the Random Forest concerning the activity recognition rate.

4.5 Discussion

In a nutshell, we showed that a physical human activity recognition system with wearable devices is feasible in a real-world scenario. However, there are technical but also conceptual aspects which we want to discuss. First, as we only considered the accelerometer but several works also proposed the gyroscope and magnetometer so other motion sensors, we think it is necessary to discuss these sensors in respect of our results. Second, even if recognizing physical activities helps in measuring physical effort or exercises it is only the tip of the iceberg in respect of supporting diabetic patients. Thus, we want propose further steps towards a more advanced activity recognition framework based on our introduced system. Third, so far we only considered classical and well know machine learning classification techniques. However, more recent and even more promising techniques are still under heavy development. This includes XGboost and LightGBM[8] which also construct trees but in a different way than Random Forest. As in our experiments the Random Forest performed the best, we consider it as necessary to discuss these classifiers and refer to them as a pointer for future work. Finally, as we focused on smart devices and an increasing number of different device types are released and so more and more different types of sensors are provided, we summarize existing works which used other than motion sensors for physical activity recognition aiming to clarify their reliability.

4.5.1 Gyroscope and Magnetometer

So far, we only considered acceleration data for recognizing physical activities where several recent publications [19, 37, 165] also often consider in addition gyration data. During our experiments, we identified three main limitations in respect of using only acceleration data. First, acceleration data is insufficient for distinguishing static activities (i.e. lying, standing, sitting) as these are characterized by very slight movements. Second, it is difficult to generalize acceleration data across people, which is an indicator that the acceleration data also covers individual information about a person which in turn is required for a satisfying recognition performance. Third, since we did not had a perfect recognition rate, other data sources would be probably helpful for improving the performance.

Most works report an improved recognition rate by relying on an accelerometer and a gyroscope which would address the third limitation; however, as a gyroscope measures the gyration of the smart device (see Section 2.2.1.2) thus movement, it is unsuitable for addressing the first limitation. Moreover, the gyration of the device usually reflects the gyration of a body part which means that the gyration data is probably also fitted to the user. Therefore, we assume that using both modalities in a cross-subjects approach would result in an even worse recognition rate. Unfortunately, we could not identify a work which focus on that problem so it can be considered as an open issue. On the other side, combining an accelerometer and a gyroscope would lead to a more accurate orientation which could help in better recognizing transitions between certain postures.

In contrast, the magnetometer gets less attention or is misinterpreted (cf. [166]) but is promising in many regards. First, combining the magnetometer with an accelerometer and gyroscope leads to a more accurate orientation estimation. Indeed, Shen et al. [165] already demonstrated that a smart-watch (providing these three sensors) is able to track the users arm. Further, as the magnetometer uses a global coordination system it is possible to compare the orientations across devices, i.e., in case that smart-devices are attached to the shin and thigh comparing their orientation may give information about the current posture (e.g. standing vs. sitting). On the other side, the orientation also can be considered as an absolute reference (cf. [62, 116]) which in turn enables to transform

[8] In 2018, both classifiers are still under active development.

the acceleration or gyration data of different devices into the same coordination system. As a consequence, dependencies or correlations can be better analyzed and identified but also device orientation changes that were affect by the sensor should no longer influence the recognition performance.

4.5.2 Sedentary Activities

So far, we only considered physical activities which allow to draw conclusions concerning physical exercises. Indeed, in respect of the treatment of diabetic patients, this is only one of several important aspects. However, the recognized physical activities are an important basis for following steps. Thus, knowing the posture allows to apply certain techniques which provide information about what is actually happening. For instance, if someone is sitting then probably the arms and the head are of most interest, i.e., in contrast to walking or running, the arms are probably moved for a certain purpose. In such a scenario, data which is gathered by a smart-watch can be combined with context information which in turn restricts the number of possible activities.

In respect of diabetes patients, sedentary activities are probably of most interest as on the one hand critical activities like intake of food or medication are mostly performed while sitting. On the other hand knowing the amount or duration of sedentary activities enables to compare the recognized physical exercises and the sedentary behavior. In fact, several works [167, 168] report that sedentary activities like watching TV may lead to an increased risk of Type 2 diabetes.

In this context, there exists several works that focus on recognizing different context information while the user is sitting. Indeed, applying such techniques while the user, e.g., is walking or lying would lead to wrong or misleading result; hence, these techniques can be considered as an extension of our proposed physical activity recognition system. For instance, Anthimopoulos et al. [169] present a vision-based food recognition system for diabetic patients for providing dietary advice through automatic carbohydrate counting. Of course, knowing when to record or interpret the video stream helps to keep the accuracy high while it also helps to protect the privacy by avoiding unnecessary video recordings. In contrast, Shen et al. [165] propose a non-video based solution using a smart-watch. They show that it is possible to recognize certain arm gestures and postures which in turn could also enable to recognize certain patterns like *fork to mouth*. Indeed, their results show that it is even possible to recognize certain wrist trajectories like writing digits or simple shapes.

Overall, we see our work as a basis for such approaches where we focused on clarifying the feasibility in a real world scenario. Thus, without a solid basis the mentioned approaches cannot be applied; hence, making the step out of the laboratory was one of our main purposes where especially we wanted to draw comparisons with existing works that usually where performed under laboratory conditions or in a limited setting. In the next chapter we will also investigate to which degree external sensors can be used for recognizing important activities and also discuss to which extend wearable and external sensors can be combined.

4.5.3 Gradient Boosting

The main causes of classification errors so that the prediction does not fit the ground truth are noise, variance, and bias. Ensemble methods try to reduce these factors by combining several classification models into one predictive model where bagging and boosting are common strategies for how to build and combine classifiers for reducing the variance or bias, respectively. Random Forest, XGBoost, and LightGBM are such ensemble methods as they consist of several decision trees where Random Forest builds bagged trees while XGBoost and LightGBM build boosted trees. The main difference between bagging and boosting is that in case of bagging the trees are built in parallel and independently (i.e. they are uncorrelated). In case of boosting, classifiers need to be built in sequence, as each classification model should learn from the errors of the preceding model aiming to minimize a loss (or cost) function. In this context, training samples are usually used to measure the performance of an individual predictor (tree) where misclassified samples gain weight and correct classified samples lose weight. This information is taken into account while the next tree is built mainly focusing on samples that were previously misclassified. Hence, the next tree always tries to recover the loss.

The Random Forest was the only classification technique, which we considered in our experiments and that is an ensemble method. As the Random Forest performed the best in each setting, we think this is evidence that ensemble methods are most suitable for physical activity recognition. In this context, several works reported across different domains that the upcoming classifiers XGBoost and LightGBM perform better which makes them highly desirable.

Even if both XGBoost and LightGBM are using boosted trees, they differ significantly especially in how the trees are created. More precisely, XGBoost uses a histogram-based algorithm for making a split decision where for each feature all values are split into discrete bins to determine the best split. In contrast, LightGBM uses a gradient-based one-side sampling strategy[9] which filters samples based on the gradient. Thus, at each node all instances having a large gradient are kept where random sampling is performed for choosing instances with small gradients. The idea is that training samples with small gradient already have a smaller training error. For comparison, the Random Forest only considers a randomly chosen subset of features at each node for making a split decision. In each case, Information Gain or Gini Index is considered for measuring a split quality.

Beside the splitting strategy, XGBoost and LightGBM also differ in respect of the growing strategy, i.e., XGBoost uses a level-wise while LightGBM uses a leaf-wise growth strategy[10]. The advantage of a level-wise strategy is to keep the tree balanced where the

[9]LightGBM also supports the histogram based algorithm but the gradient-based one-side sampling strategy is provided by LightGBM exclusively.

[10]In recent implementations, XGBoost also supports the leaf-wise growth strategy.

leaf-wise strategy can produce very deep branches which in turn makes it more prone to overfitting. However, the advantage of the leaf-wise strategy is to be more flexible where the result of a leaf-wise strategy can be the same as of a level-wise strategy but not vice versa. In this context, the leaf-wise strategy chooses always the node which reduces the loss the most.

We want to highlight that XGBoost and LightGBM are not the only implementations of gradient boosting decision trees but are the most promising once. As the performance of classifiers is inherently data dependent, it is not clear if they perform in respect of motion sensor data and physical activity recognition. Therefore, we refer to these classifiers as future work.

4.5.4 Beyond Motion Sensors

Accelerometer, Gyroscope, and Magnetometer are just three out of several sensors which are nowadays provided by smart devices and that got most attention in respect of physical activity recognition. Indeed, physiological signal or vital signs but also environmental variables may sound promising but most works tend to report misleading or inaccurate results. In the following, we want to provide an overview of sensors which were considered for physical activity recognition where we summarize the reported results for clarifying the opportunities and limitations.

EKG [23, 170] An EKG (or heart rate) sensor is usually combined with an accelerometer aiming to recognize physical effort. In this context, Juha et al. show that it is possible to distinguish between different levels of walking (i.e. speed) but they also state that especially activities of short duration such as climbing stairs leads to a classification errors across all considered activities. Pärkkä et al. state that this can be attributed to the fact that the heart rate reacts to activity changes with a delay so a person might be already standing or sitting for a while where the heart rate is still increased. Further, even when the heart rate correlates with the intensity level, it seems to be difficult to distinguish between certain types of activities (e.g. walking vs. cycling).

GPS [171–173] A GPS sensor is commonly used to determine the current location where this sensor is restricted to an outdoor scenario as it is only able to recognize the location under the open sky. For that reason, several works suggested to combine a GPS sensor with an acceleration sensor to benefit from it whenever possible. In this context, most works focus on the user's speed and try to distinguish between walking and non-walking or certain mobility modes. Reddy et al. clarifies by computing the information gain that the GPS speed is a valuable feature even when using acceleration based features. Thiagarajan et al. come to the same conclusion but highlight that using only a GPS sensor does not enable to distinguish between different types of movement having roughly the same movement speed (e.g. walking, running, and jogging).

Pressure [102, 174] A barometric or pressure sensor measures the pressure of the air and is currently only provided by few smart devices. Indoor navigation is probably one of the most interesting scenarios as the pressure might provide information about the user's current floor level and so in turn might be helpful to avoid confusion between walking and climbing stairs. In this context, Muralidhara et al. report a very high accuracy for recognizing if someone is standing on an escalator or in an elevator or is climbing up the stairs. Further, as the accelerometer is usually the first choice they also compare these two sensors in respect of robustness (i.e. the smart-phone is in use). They report that the recognition performance stays high when using the pressure sensor while it drops significant using the accelerometer. Besides, they highlight that the absolute pressure values have significant time-of-day variations while the change (delta) is remarkably consistent and steady for any given building. On top, the pressure sensor is robust to changes in the on-body device position and the orientation.

Microphone [102, 175] While a microphone might provide valuable information about the current environment, it can be also helpful in recognizing sitting or standing while the body is accelerated due to being in a bus or subway. In this context, Han et al. show that it is feasible to distinguish between several different place by analyzing the audio data and that this in turn enables to optimize the considered set of features. They report a significant improvement in respect of the classification accuracy for ambulatory activities but also certain transportation modes. A part from that, Khan et al. combined an accelerometer and a microphone for recognizing 15 activities (physical activities but also ADLs). Their results show that the audio data contribute to the overall recognition performance but it is unclear in respect of the individual activities.

Wi-Fi [176] Wi-Fi signal based activity recognition systems rely on the channel state information which comprises properties of the communication link including scattering, fading, and power decay with distance. Indeed, such a system is restricted to a certain environment, as it requires, in addition to a smart-device, also a Wi-Fi access point. The idea is to measure changes of the mentioned properties to estimate the speed of the user to determine if the user for example is running, walking, or sitting. In this context, Wang et al. clarify that it is possible to detect both high-speed movement and low-speed movement. This includes short actions such as boxing, falling and common physical activities like walking or running. However, they conclude that multiple people within the same room lead to signal interferences. Thus while it is still possible to recognize activities if just on person is moving, multiple Wi-Fi access points are required to handle the movement of several people.

EEG [177, 178] An electroencephalography (EEG) records electrical activities of the brain. Indeed, at present this sensor is usually not part of a smart-phone or any other widespread smart device. However, smart-headgears such as smart-headbands (e.g. *BrainPlus: Smart EEG Device for Your Better Brain 2*[11]) are on the move. Diambra et al. already demonstrated in 1990 that an EEG enables to recognize epileptic activities. Nowadays, activities can be even more granular recognized. Zhang et al. show that it is possible to recognize the user's intention in respect of closing eyes, moving hands (left, right, or both), and using feet. However, multi-class classification is a major challenge in respect of EEG signals. Zhang et al. state that most existing works focus on binary classification as existing approaches usually have an inferior performance in a multi-class setting.

Video [179–181] Be it first-person or third-person view, using visual information for activity recognition is an active field of research. While most works focus on recognizing ADLs, there are also works which focus on physical activities. For example, Zhan et al. show that the optical flow of the first-person view can be used for recognizing and distinguishing between walking and climbing stairs up or down. Ballin et al. use the same technique but in respect of a third-person view. They record and analyze depth images and transfer the recognized movements in a 3D grid to derive the actual activity. Of course, there are many other techniques such as analyzing silhouettes, interpreting visible body parts, considering spatiotemporal features, detecting occupancy patterns, and recognizing active objects. Moreover, several works combine video and motion sensors to compensate certain drawbacks. These include privacy issues, the fact that the camera not always captures the scene of interest, and in case of third-person view the restriction to a certain location. However, discussing these works would be out of scope; hence, we would like to refer the reader to the following works [182–185].

Of course, there are even more sensors such as proximity, humidity, skin temperature and light. However, to the best of our knowledge, there is no study which investigates these sensors for recognizing physical activities.

[11]`https://www.brainplus.co`, last access 06.12.18

Chapter 5

Activity Recognition within Smart Environments

In this chapter, we focus on the introduced open issues in respect of recognizing Activities of Daily Living (see Sections 1.2 and 1.3). Overall, we aim to deploy a reliable and feasible recognition system which overcomes common limitations of existing system. For that reason, first we introduce a basic concept which is subsequently enhanced by online recognition and active learning components. Finally, we discuss a combination with the physical activity recognition approach (see Chapter 4) to clarify the advantages and opportunities.

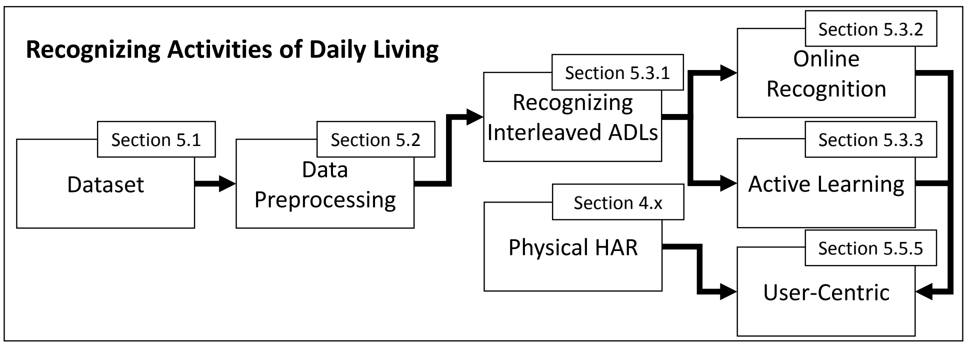


Figure 5.1: Recognizing Activities of Daily Living in a Smart-Environment

For that purpose, first we introduce two datasets which we use to evaluate our approaches (Section 5.1, published in [14,17]). Subsequently, as in the preceding chapter, we explain the required preprocessing steps for the data handling but also for improving the quality concerning irrelevant and redundant information (Section 5.2, published in [2]).

Then, we describe the basic concept of our recognition system where we aim on the one hand to clarify the performance and feasibility of a probabilistic and ontology based system and on the other hand to use it afterwards to evaluate also extensions (Section 5.3.1, published in [2]). This includes online recognition (Section 5.3.2, published in [8]) and active learning (Section 5.3.3, published in [5]), i.e., recognizing the ADLs at almost real-time while adapting the model to the current situation. Please see Appendix A for further details regarding the contribution of the individual authors.

5.1 Activities of Daily Living Datasets

In contrast to physical human activity recognition, in the following we introduce and use third-party datasets to answer our initial research question but also to investigate related issues. In general, both datasets describe signals of a sensor network in a smart-environment which were triggered due to certain actions of people. In particular, the first dataset (CASAS, Section 5.1.1) stands out due to the size (i.e. number of considered people and activities) while the second dataset (SmartFABER, Section 5.1.2) was recorded in a fully naturalistic environment. Please not that we focus only on a single resident scenario, i.e., there is at most one person in the smart-home.

5.1.1 CASAS: A Smart-Home in a Box

The CASAS dataset was recorded and published by G. Singla, D.J. Cook, and et al. [17, 186]. They equipped a common living room and a kitchen with 42 sensors to gather the location of the resident, the usage of doors, the interaction with certain items, and taking water from the faucet (see Figure 5.2 and Table 5.1). The door sensors (D_{xy}) recognize the *opening* and *closing* event where contact sensors (I_{xy}) only gather if an item is present in a predefined location. In addition, the movement sensors (M_{xy}) record the entering and leaving in the range of operation.

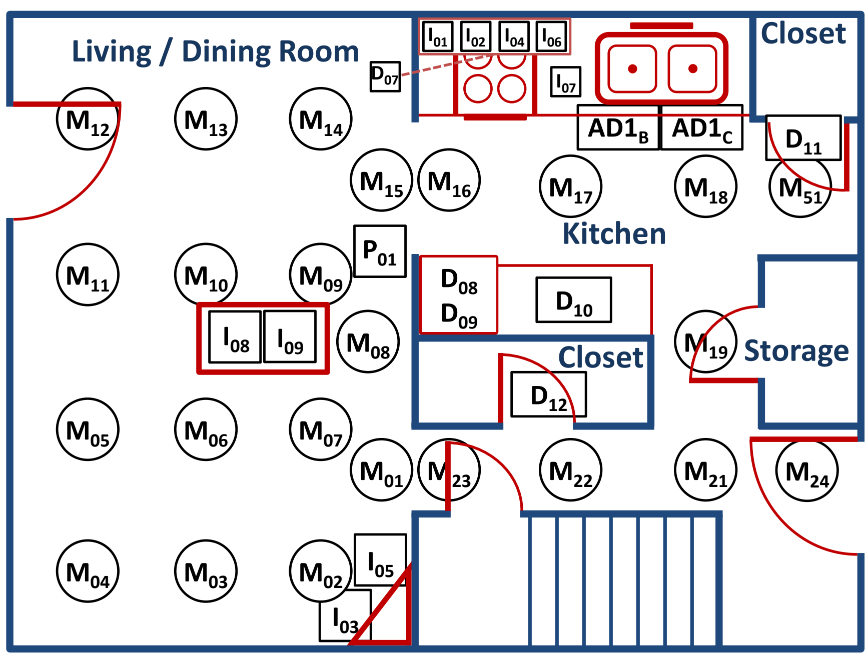


Figure 5.2: Smart-home apartment and sensor locations (adapted from [186]).

During the data collection, only one single person was present in the smart-home. Each was introduced, first, to perform eight predefined ADLs in a certain order. Subsequently, they had to repeat these activities but interweaving with the goal of being efficient so the order and expenditure of time were up to the subject. As an illustration, Figure 5.3 shows a resident taking water from the faucet in the kitchen. In the following, we outline the performed activities in detail:

Fill medication dispenser (ac_1) - The resident has to refill a medication dispenser. Both, the drugs and the dispenser are located in the wall cupboard (D_{07}, I_{04}, I_{06})

in the kitchen. In this context, the free space between the oven and the sink on the kitchen counter is used to refill. There was no instruction to put it back.
(Avg. duration: 3.5 minutes, avg. number of sensor events: 31)

Watch DVD (ac_2) - The resident takes a DVD from the TV shelf (I_{03} and I_{05}) in the living room to watch it. The DVD player and the TV are located on top of the shelf. After watching it, the resident has to turn off the TV and has to put the DVD to the original place.
(Avg. duration: 7 minutes, avg. number of sensor events: 59)

Water plants (ac_3) - The resident has to water three plants which are located in the living room (living room table, next to I_{08} and I_{09}) and in the kitchen (next to the closet, D_{11}). For that, the resident has to take the water can which is located in the closet (D_{11}), then fill it with water ($AD1_B$, $AD1_C$), and subsequently move to the plants to water them. Finally, the can is emptied into the sink (kitchen) and put back in the closet.
(Avg. duration: 1.5 minutes, avg. number of sensor events: 71)

Table 5.1: Description of the sensors that were used and recorded in the smart-home.

Sensor ID	Sensor Type	Sensor Location	Signal	Description
M_{01},...,M_{51}	presence	*everywhere*	binary	to capture movement
I_{01}, I_{02}, I_{04}, I_{06}	contact	kitchen	binary	shelves of the wall cupboard
I_{03}, I_{05}	contact	living room	binary	right and left TV shelves
I_{07}	contact	kitchen	binary	pot sensor
I_{08}	contact	living room	binary	phone book sensor
I_{09}	contact	living room	binary	birthday card sensor
D_{07}	magnetic	kitchen	binary	door of the wall cupboard
D_{08}, D_{09}, D_{10}	magnetic	kitchen	binary	freezer, fridge, and microwave door
D_{11}, D_{12}	magnetic	kitchen	binary	storage door
$AD1_B$, $AD1_C$	water	kitchen	number	taking hot or cold water
P_{01}	contact	living room	binary	touching the phone

Figure 5.3: In relation to Figure 5.2, the image depicts the area around M_{17} and M_{18} (adapted from [186]).

Answer the phone (ac_4) - The resident has to move to the phone when it rings. The phone is located in the living room (P_{01}), close to the kitchen. The conversation includes several question which are answered. Afterwards, the residents just hangs up.
(Avg. duration: 2 minutes, avg. number of sensor events: 31)

Prepare birthday card (ac_5) - The resident has to move to the living room table to prepare a birthday card (next to I_{08} and I_{09}). All required items are located on the table. First, the resident writes an appropriate text into the birthday card (I_{09}) and fills out a check as a birthday gift. Subsequently, both are put in an envelope and an address it written on it using the address book (I_{08}).
(Avg. duration: 4 minutes, avg. number of sensor events: 56)

Prepare soup (ac_6) - The resident has to prepare a noodle soup in the kitchen. The required ingredients are located in the wall cupboard (D_{07}, I_{01} and I_{02}) and the bowl is located on the counter (I_{07}). In addition to the ingredients, the resident has to use the water pitcher that is located in the refrigerator (D_{09}). If necessary, the water faucet is used ($AD1_B$, $AD1_C$). If desired, the soup is heated by the microwave (D_{10}).
(Avg. duration: 5.5 minutes, avg. number of sensor events: 96)

Clean (ac_7) - The resident has to sweep the kitchen floor and to dust the living room. The ADL is not bound to a certain location and also the order and duration is not specified. The required supplies are located in the kitchen closet (D_{11}).
(Avg. duration: 4 minutes, avg. number of sensor events: 118)

Choose outfit (ac_8) - The resident has to walk to the clothes closet (D_{11}) to choose an outfit for a job interview. Then, the resident has to carry the chosen outfit to the living room couch (close to the TV, I_{03} and I_{05}).
(Avg. duration: 1.5 minutes, avg. number of sensor events: 34)

These activities were chosen with respect to clinical questionnaires [187]. The recorded data was manually annotated.

5.1.2 SmartFABER

The SmartFABER dataset was created by Riboni et al. [14, 188] and has been acquired during three months. In contrast to the CASAS dataset, this dataset was recorded in a fully naturalistic environment. Hence, the resident was an elderly woman aged 74, living alone, and diagnosed with mild cognitive impairment and medical comorbidities. In this context, she had to take three different medication, two in the morning and the remaining one in the evening. She was observed for 55 days in her kitchen (see Figure 5.4) focusing on *preparing meal*, *eating*, and *taking medicines*. As she was observed during her daily routine, there were no instruction how these activities have to be performed. Indeed, due to her cognitive decline, the activities have been performed in many different and sometimes unexpected ways. Hence, the recognition of those ADLs is challenging, even if the number of considered activities is limited.

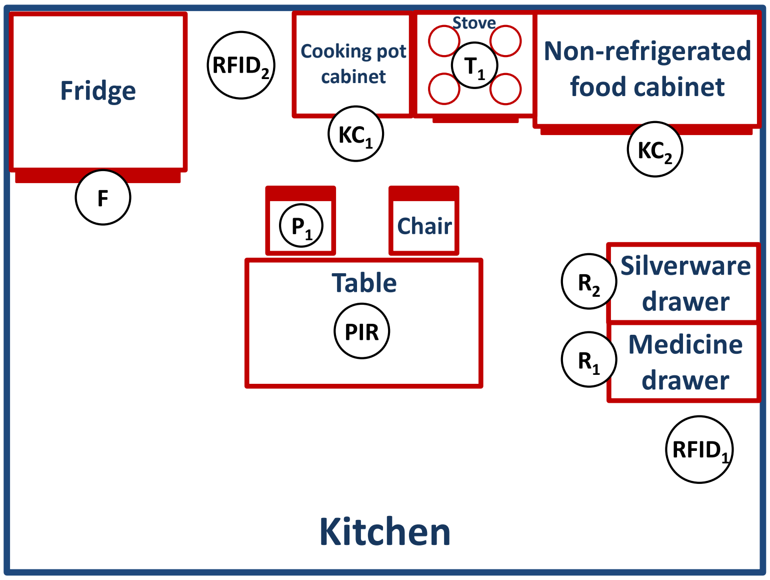


Figure 5.4: Smart-home apartment and sensor locations (sketch, adapted from [188]). In addition, there are three RFID tags that are attached to three different medications and 15 tags as cards for certain food items.

For the observation, the kitchen was equipped with 10 sensors including presence, pressure, magnetic, and temperature sensors. In addition, RFID tags were attached to 15 food items like fish, potatoes or rice but also to the three medicine boxes. More precisely, in case of the food items, there were related cards with the corresponding tags and in general, the resident had to swipe the RFID tags near an RFID reader. Of course, in respect of the diagnosed disease there was no guarantee that the elderly woman used the RFID tags in the desired way. Besides, the RFID reader was also not very reliable. Apart from that, the acquired data is also affected by noise due to various technical issues encountered during data acquisition.

Table 5.2: Description of the sensors that were used and recorded in the smart-home.

Sensor ID	Sensor Type	Sensor Location	Signal
P_1	pressure	on the chair	binary
PIR	presence	above the dining table	binary
T_1	temperature	above the stove	numeric
KC_1, KC_2	magnetic	kitchen cabinets	binary
R_1, R_2	magnetic	drawer/repositories	binary
F	magnetic	freezer	binary
$RFID_1$, $RFID_2$	RFID reader	attached to the wall	nominal

Table 5.2 summarizes the deployed sensors and their characteristics. Concerning the temperature sensor, a threshold of 29°C was set to ascertain if the stove is in use. At the end of each day, the recorded data was transmitted and subsequently manually annotated.

As the resident also performed other activities, the sensor events were in addition to *Taking medicines* (ac_9), *Cooking* (ac_{10}), and *Eating* (ac_{11}) also annotated with *Others* (ac_{12}). The dataset is not publicly available.

5.2 Data Preprocessing

Compared to the physical activity recognition dataset, the effort concerning the preprocessing of the CASAS and SmartFABER datasets is significantly less. That is mainly because the raw sensor signals were already transformed into states, e.g., a motion sensor is associated with on/off while an item interaction sensor is associated with absent/ present. Further, in the beginning we do not segment the sensor data into windows but try to interpret all sensor data of a complete day at once (i.e. offline activity recognition). Thus, in the following section, we summaries cleaning and editing steps (Section 5.2.1) while the window segmentation techniques (Section 5.2.2) refer to how we segment the sensor events considering an online activity recognition scenario which we investigate subsequently.

5.2.1 Data Cleaning and Editing

The CASAS dataset[1] (#3, Interweaved ADL Activities) consists of separate files for each resident where in turn, for each resident exists several files which describe sequential or interleaved performed activities. For our experiments, we only considered the files which describe *interwoven* (interleaved) activities. While the dataset is not free from (sensor) errors (e.g. one prepared a soup without water), we decided to use the dataset basically as it is. In particular, we only modified the following things:

- We excluded resident *p22*, as the records were incomplete (i.e. only sequential recordings were available).
- We removed sensor event *E01* as it only occurred in respect of resident *p17*. The meaning of *E01* was also unclear as there was no description available.
- We removed sensor event *M26* as it only occurred in respect of resident *p04*. Further, the location of this sensor was also not clear.

The SmartFABER dataset was already revised by the original authors [14]. For that reason, we decided to use the dataset as it is. Not least to be comparable to already published results.

5.2.2 Window Segmentation Techniques

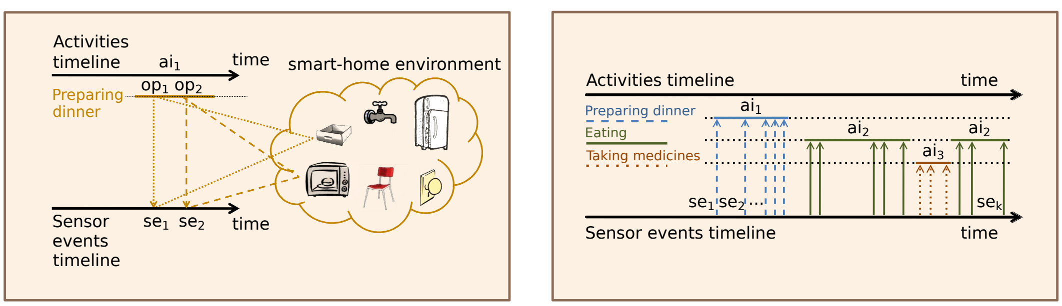
As we focus on recognizing ADLs in offline but also online mode, we rely on different segmentation strategies. For offline recognition, we collect the sensor events of an entire

[1] http://casas.wsu.edu/datasets, last access: 14.12.2018

day and subsequently analyze and interpret them. Thus, the sensor events are actually not segmented but directly labeled by the activity that most probably generated them. In regard of online recognition, it is not feasible to start the recognition process at the end of the day. For that purpose, we draw on the already introduced static and dynamic windowing approaches (see Figure 4.6). As a baseline, we consider static, overlapping windows where the window length is not defined by time but by a fixed number of sensor events. The actual idea is to investigate dynamic, overlapping windows, i.e. to find suitable rules or patterns on which we can rely to decide how long the respective window should be. Compared to our introduced physical activity recognition approach where we used static windows, the sensor events which are provided by a smart-home network are less abstract and less noise. Indeed, each sensor event can be associated with a certain meaning. This is why we believe that dynamic windows are more appropriated than static windows. Besides, as soon as a new dynamic window is finalized (i.e. completed), it is associated and analyzed in regard of the preceding windows. Thus, in a post-processing step certain windows might be summarized as they describe the same activity; however, we want highlight that summarized windows not necessarily need to be consecutive windows. We outline further details in the subsequent sections. In the following, we use the term *segment* instead of *window* just to be in line with the existing literature. Actually, these two terms can be considered as synonyms but for some reason it is common to use the term *window* in respect of physical activity recognition while the term *segment* seems to be preferred in regard of recognizing ADLs.

5.3 Methods

We assume a smart-home instrumented with sensors to detect interactions with items and furniture, context conditions (e.g., temperature), and presence in certain locations. Further, we assume that there is only a single resident within the smart-home. We denote



(a) Time-lines

(b) Sensor events

Figure 5.5: Connection between performed activities, resulting operations, and triggered sensor events. The time-lines (left) illustrate that the sensor network records certain operations (e.g. item usage) of an executed activity. Subsequently, these sensor events are used to reconstruct the activity instances that generated these sensor events (right). The individual sensor events, their relations, and dependencies indicate by which activity they were generated.

by *activity class* (ac) an abstract activity (e.g., cooking and cleaning), and by *activity instance* (ai) the actual occurrence of an activity of a given class during a certain time period. In this context, we consider $\mathbf{A} = \{ac_1, ac_2, \ldots, ac_k\}$ as the set of activity classes and an instance ai_i of an activity class $ac_j \in \mathbf{A}$ represents the occurrence of ac_j during a given timespan. The activity instance is associated to the operations executed to perform it, where the start and end time of instances of different activities can overlap. Figure 5.5a illustrates the relation between recorded sensor events and an activity instance. Hence, during the execution of activity instance ai_1 (*preparing dinner*), the subject executes the operations op_1 (opening the silverware drawer) and op_2 (turning on the microwave oven). Supposing that sensors are available to detect these operations, op_1 and op_2 generate two sensor events se_1 and se_2, whose timestamp corresponds to the time of the respective operation.

Based on the observation of a set of timestamped sensor events, the goal of the activity recognition system is to reconstruct the most probable activity instances that generated them. As shown in Figure 5.5b, we achieve this goal by assigning each event se_i to the activity instance that most probably generated it. This approach allows us to recognize interleaved activities, as it is the case for ai_2 and ai_3 (the subject temporarily interrupts the meal to take medicines). In the following, we introduce a system that overcomes several limitations of existing systems by implementing the mentioned concept, still focusing on an approach that is applicable in a real world scenario.

5.3.1 Recognizing Interleaved Activities of Daily Living

ADL recognition techniques are divided into two categories: *data-driven* and *knowledge-based*. The former is based on supervised learning while the latter exploits logic formalisms (e.g., ontologies) to represent formally sensor events and activities. In order to combine the strength points of both approaches, we rely on Markov Logic Networks (MLN) [2,189–191] (see Section 2.5).

Figure 5.6 depicts an overview of our system. Hence, we used an OWL 2 ontology [142] which formally models a smart-home environment and the semantics of activities (see Section 2.4). We rely on ontological reasoning to derive necessary conditions about the sensor events that must occur during the execution of a specific activity in the current environment. This also enables to extract *semantic correlations* among triggered sensor events and performed ADLs. Using this information, probabilistic reasoning derives the activity that most likely generated the recorded sensor events. More precisely, the SEMANTIC CORRELATION REASONER performs ontological reasoning to derive semantic correlations among event types and activity classes; e.g., "the event type *UseStove* is strongly related to *PreparingHotMeal* and unrelated to *PreparingColdMeal*". Those correlations are used by the module for STATISTICAL ANALYSIS OF EVENTS to identify *candidate* activity instances. These are initial hypotheses about the start and end time of occurred activities. Subsequently, the events of the sensor network and these candidates are used to

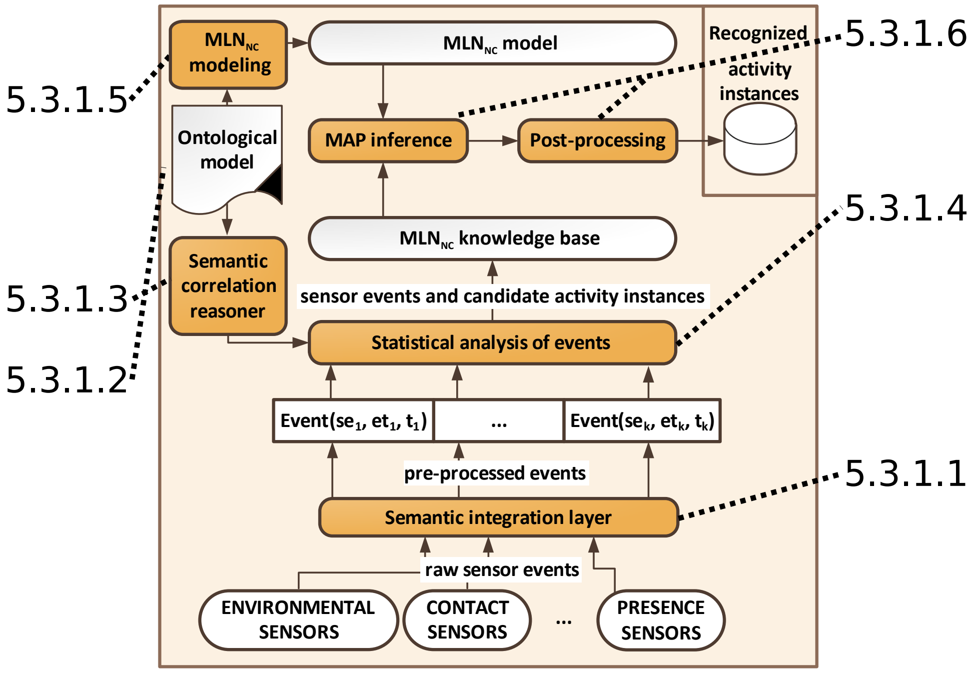


Figure 5.6: System overview. The *statistical analysis* layer combines the information received from the sensors and the ontological model to build a *knowledge base*. *MAP inference* enables to derive the most probable world from this knowledge base considering the MLN_{NC} *model*. This results in the recognition of the actual activity instances.

populate the assertional part of the MLN_{NC} knowledge base. Simultaneously, the ontological model of considered activities and events is translated into the MLN_{NC} model. Periodically (e.g., at the end of each day), MAP INFERENCE is performed to assign each event to the candidate activity instance that most probably generated it, according to semantic correlations and ontological constraints. Finally, the output of MAP INFERENCE is POST-PROCESSED to detect the exact start and end time of occurred activity instances.

In the following, we go into detail and explain the concepts and functionality of these components (see Figure 5.6).

5.3.1.1 Semantic Integration Layer

The smart-home monitoring system collects raw events data from the sensor network, including environmental, presence, and contact sensors. The SEMANTIC INTEGRATION LAYER applies simple pre-processing rules to detect operations from raw sensor events. For example, if at time t the fridge door sensor produces the raw event *open*, then the operation at t is *opening the fridge*. We denote $\mathbf{E}$ as the set of pre-processed event types that correspond to the set of monitored operations (e.g., $\mathbf{E} = \{$ *opening_the_fridge*, *closing_the_fridge* $\}$). In addition, $\mathbf{T}$ describes the set of all possible event timestamps. A temporally ordered set of events is represented as follows:

$$\langle\, Event(se_1, et_1, t_1), \ldots, Event(se_k, et_k, t_k)\,\rangle$$

where $Event(se_i, et_i, t_i)$ indicates that se_i is an instance of the event type $et_i \in \mathbf{E}$ occurred at timestamp $t_i \in \mathbf{T}$. This set of events is forwarded to the STATISTICAL ANALYSIS OF EVENTS layer for segmentation followed by analyzing.

5.3.1.2 Ontological Model

As a basis, we reused an OWL 2 ontology of a related work [142] which defines the semantics of activities and operations (see Section 2.4). Figure 5.7 illustrates an excerpt of this ontology, which describes a complete home environment. In addition, it also covers axioms for each activity class that describe dependencies and conditions. In particular, we express necessary conditions for a set of operations to be generated by an instance of that class, according to the activity semantics. For example, the operations generated by an instance of *preparing hot meal* must include an operation *using a cooking instrument*. In this context, the ontology also covers sensor classes and corresponding operations that they detect; e.g., a power sensor attached to the electric stove detects the operation *turning on the stove*. In turn, this operation is a subclass of *using a cooking instrument*. The ontology carefully describes these kinds of relations and enables to derive certain constraints through ontological reasoning. For instance, "Since the stove is the only cooking instrument in the home, and a sensor is available that detects the usage of the stove, then each instance of *preparing hot meal* executed in the home must necessarily generate an event from that sensor".

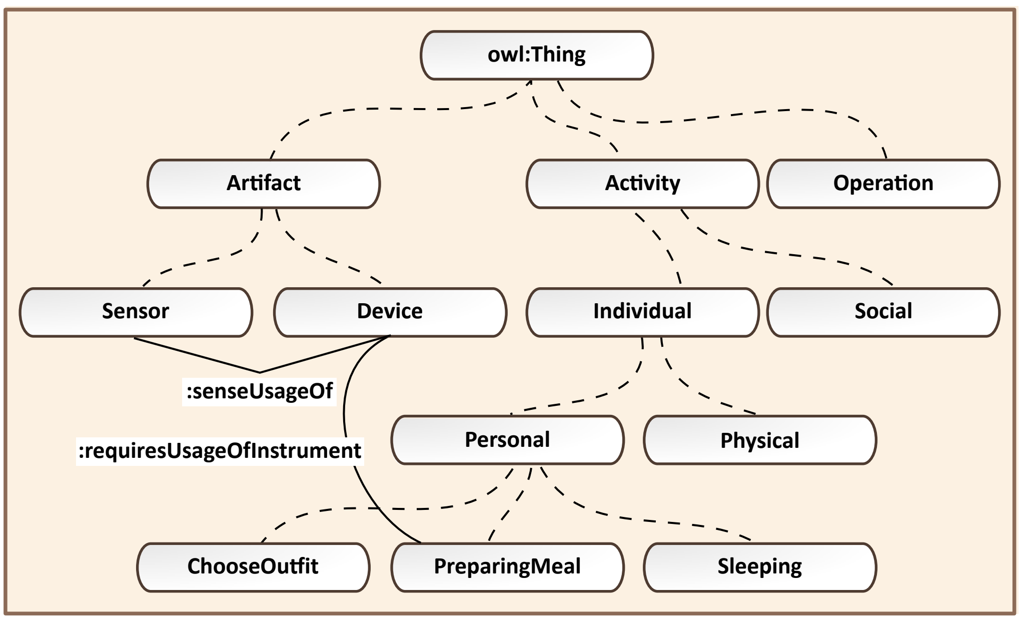


Figure 5.7: Excerpt of the ontology. The dashed lines represent a *subClassOf* relation where the upper is the parent of the lower class. In addition, the individual classes have relations that describe dependencies.

In addition to activity and object correlations, we also take time and location dependencies into account. This includes constraints on the duration of the activity instance and the relation between an activity and a certain location. In the following, we explain how we use ontological reasoning to infer these probabilistic dependencies among sensor event types and classes of executed activities; we denote them as *semantic correlations*. The ontology is publicly available[2].

5.3.1.3 Semantic Correlation Reasoner

We rely on ontological reasoning to mine semantic correlations among event types and activity classes, and to derive necessary conditions about the sensor events that must occur during the execution of specific activity instances in the current environment. In the following, we introduce a simple running example to illustrate our approach.

> **Example 2** *Suppose to monitor three activities in a smart home: preparing hot meal, preparing cold meal, and preparing tea. The home contains one silverware drawer, one stove, and one freezer, each equipped with a sensor to detect its usage. No training set of activities is available. How can we exploit semantic reasoning to recognize the activities?*

In the following of this section, we explain how we answer the above question. The specific objective of this reasoner is to compute the degree of correlation among sensor events and the ADLs performed in the home. As illustrated in the axioms below, in the ontology, artifacts are organized in a hierarchy. The class Stove is a sub-class of cooking instruments, used in the apartment to prepare hot meal or tea, where Freezer is a Device used to prepare hot or cold meal. SilverwareDrawer belongs to FoodPrepFurniture and is required by all three activities. The instance {apt} represents the current apartment. For clarification, we represent the name of ontological instances within curly brackets.

$$\textsc{Stove} \sqsubseteq \textsc{CookingInstrument} \sqcap \Big(\exists\, \textsc{usedFor}.\big((\textsc{PrepHotMeal} \sqcup \textsc{PrepTea}) \sqcap (\exists\, \textsc{occursIn}.\{\textsc{apt}\})\big)\Big).$$

$$\textsc{Freezer} \sqsubseteq \textsc{Device} \sqcap \Big(\exists\, \textsc{usedFor}.\big((\textsc{PrepHotMeal} \sqcup \textsc{PrepColdMeal}) \sqcap (\exists\, \textsc{occursIn}.\{\textsc{apt}\})\big)\Big).$$

$$\textsc{SilverwareDrawer} \sqsubseteq \textsc{FoodPrepFurniture}.$$

$$\textsc{FoodPrepFurniture} \sqsubseteq \textsc{Furniture} \sqcap \Big(\exists\, \textsc{usedFor}.\, \big((\textsc{PrepTea} \sqcup \textsc{PrepColdMeal} \sqcup \textsc{PrepHotMeal}) \sqcap (\exists\, \textsc{occursIn}.\{\textsc{apt}\})\big)\Big).$$

[2] `https://sensor.informatik.uni-mannheim.de/#results2016unsupervised`

Based on the smart-home setup, we instantiate the ontology with the sensors and artifacts in the apartment, and we specify which activities we want to monitor.

Example 3 *The activities that we want to monitor are* $\{\textsc{ac\_prep\_cold\_meal}\}$, $\{\textsc{ac\_prep\_hot\_meal}\}$ *and* $\{\textsc{ac\_ prep\_tea}\}$. *They are instances representing the generic occurrences of* PrepColdMeal, PrepHotMeal, *and* PrepTea, *respectively. Lines 5.5-5.7 state that at most one instance of each activity type can be monitored at a time. Further, lines 5.8-5.10 represent that the* $\{\textsc{apt}\}$ *contains exactly one cooking instrument, one silverware drawer, and a freezer:*

$$\{\textsc{apt}\} = \textsc{Apartment} \tag{5.1}$$
$$\sqcap\, (\exists\, \textsc{monitAct}.(\{\textsc{ac\_prep\_cold\_meal}\})) \tag{5.2}$$
$$\sqcap\, (\exists\, \textsc{monitAct}.(\{\textsc{ac\_prep\_hot\_meal}\})) \tag{5.3}$$
$$\sqcap\, (\exists\, \textsc{monitAct}.(\{\textsc{ac\_prep\_tea}\})) \tag{5.4}$$
$$\sqcap\, (\leq\ 1\textsc{monitAct}.\textsc{PrepColdMeal}) \tag{5.5}$$
$$\sqcap\, (\leq\ 1\textsc{monitAct}.\textsc{PrepHotMeal}) \tag{5.6}$$
$$\sqcap\, (\leq\ 1\textsc{monitAct}.\textsc{PrepTea}) \tag{5.7}$$
$$\sqcap\, (=\ 1(\textsc{isIn})^{-}.\textsc{CookingInstrument}) \tag{5.8}$$
$$\sqcap\, (=\ 1(\textsc{isIn})^{-}.\textsc{SilverwareDrawer}) \tag{5.9}$$
$$\sqcap\, (=\ 1(\textsc{isIn})^{-}.\textsc{Freezer}). \tag{5.10}$$

Subsequently, we introduce an instance in the ontology for each artifact in the apartment:

$$\{\textsc{stove}\} \equiv \textsc{Stove} \sqcap \exists\, \textsc{isIn}.\{\textsc{apt}\}.$$
$$\{\textsc{freezer}\} \equiv \textsc{Freezer} \sqcap \exists\, \textsc{isIn}.\{\textsc{apt}\}.$$
$$\{\textsc{silverware\_drawer}\} \equiv \textsc{SilverwareDrawer} \sqcap \exists\, \textsc{isIn}.\{\textsc{apt}\}.$$

We also instantiate each sensor that occurs in our apartment:

$$\{\textsc{s\_stove}\} \equiv \textsc{PowerSensor} \sqcap (\exists\, \textsc{sensesUsageOf}.\{\textsc{stove}\}) \sqcap (\exists\, \textsc{producesEvent}.\{\textsc{et\_stove}\}).$$

$$\{\textsc{s\_silverware\_drawer}\} \equiv \textsc{ContactSensor} \sqcap (\exists\, \textsc{sensesUsageOf}.\{\textsc{silverware\_drawer}\}) \sqcap (\exists\, \textsc{producesEvent}.\{\textsc{et\_silverware\_drawer}\}).$$

$$\{\textsc{s\_freezer}\} \equiv \textsc{ContactSensor} \sqcap (\exists\, \textsc{sensesUsageOf}.\{\textsc{freezer}\}) \sqcap (\exists\, \textsc{producesEvent}.\{\textsc{et\_freezer}\}).$$

According to the introduced axioms, $\{\textsc{s\_stove}\}$ is an instance of PowerSensor that senses the usage of $\{\textsc{stove}\}$ and produces a generic event of type $\{\textsc{et\_stove}\}$. Similarly, the last two axioms define sensors and events for the silverware drawer and the freezer, respectively.

We exploit the property composition operator to infer the semantic correlations between sensor events and activity types. In particular, we use the following axiom, which states that: "if an event of type *et* is produced by a sensor that detects the usage of an artifact possibly used for an activity of class *ac*, then *et* is a *predictive sensor event type* for *ac*":

$$\text{PRODUCESEVENT}^{-} \circ \text{SENSESUSAGEOF} \circ \text{USEDFOR} \rightarrow \text{PREDICTIVESENSOREVENTFOR}$$

Then, we perform ontological reasoning to infer the fillers of property PREDICTIVESENSOREVENTFOR, and use them to compute semantic correlations.

Example 4 *Considering all of the introduced axioms, the OWL 2 reasoner infers that:*

- {ET_STOVE} *is a predictive sensor event type for* {AC_PREPARE_HOT_MEAL} *and* {AC_PREP_TEA}.
- {ET_SILVERWARE_DRAWER} *is a predictive sensor event type for* {AC_PREP_HOT_MEAL},{AC_PREP_COLD_MEAL } *and* {AC_PREP_TEA}.
- {ET_FREEZER} *is a predictive sensor event type for* {AC_PREP_HOT_MEAL} *and* {AC_PREP_COLD_ MEAL}.

We represent semantic correlations using a *prior probability matrix* (PPM). The rows correspond to the activity classes, while the columns to the sensor event types. Hence, $PPM(ac, et)$ stores the probability of an event of type *et* being generated by an activity of class *ac*. If a given sensor event type is predictive of a single activity class, the value of the corresponding entry is one; if it is predictive of multiple activity classes, the value is uniformly distributed among them. The prior probability matrix resulting from our running example is shown in Table 5.3. The PPM is given as input to the STATISTICAL ANALYSIS OF EVENTS layer.

Table 5.3: Prior probability matrix of our running example.

	et_stove	**et_silverware_drawer**	**et_freezer**
ac_prep_hot_meal	0.5	$0.\overline{33}$	0.5
ac_prep_cold_meal	0.0	$0.\overline{33}$	0.5
ac_prep_tea	0.5	$0.\overline{33}$	0.0

5.3.1.4 Statistical Analysis of Events

Both, the results of the SEMANTIC INTEGRATION LAYER and the SEMANTIC CORRELATION REASONER are required by the STATISTICAL ANALYSIS OF EVENTS layer, i.e., the prior probability matrix and the preprocessed sensor events. Using this, we identify activity instance candidates and consider them in addition to the observed sensor events and the computed semantic correlations as part of our MLN_{NC} knowledge base. Candidate

Algo 3 Statistical analysis of events

Input: sensor events
$X = \{ev(se_0, et_0, t_0), \ldots, ev(se_n, et_n, t_n)\}$,
prior probability matrix PPM
Output: candidate activity instances $\{ai_0, ai_1, \ldots, ai_{m-1}\}$

$instances \leftarrow \emptyset$
for each $ev(se, et, t) \in X$ **do**
  $ac \leftarrow$ activity class with max correlation with et according to PPM
  $ai \leftarrow$ activity instance in $instances$ of class ac closest to se
  if ai exists **and** t is temporally close to ai according to $maxDelay_{ac}$ **then**
    assign $ev(se, et, t)$ to ai
  else
    $ai \leftarrow$ a new instance of class ac
    assign $ev(se, et, t)$ to ai
    $instances \leftarrow instances \bigcup \{ai\}$
  end if
end for
return $instances$

activity instances are computed by a heuristic algorithm (see Algorithm 3) which implements the STATISTICAL ANALYSIS OF EVENTS module. The algorithm iterates over all temporally ordered events and simultaneously uses the PPM of semantic correlations to infer, for each sensor event se, the most probable activity class ac generating it. The corresponding timestamp of the event and the resulting activity class enables us to formulate initial hypotheses about the occurred activity instances. If an activity instance ai of class ac exists, whose boundaries (start and end time) are temporally close to se according to an activity-dependent threshold $maxDelay_{ac}$, then se is assigned to ai. Otherwise, a new instance of class ac is created, and se is assigned to it. The boundaries of each instance are respectively represented by the first and the last event of the instance.

Then, MAP inference enables us to assign each activity instance to its most probable class, and each event to its most probable activity instance. For that, we introduce in the following the corresponding MLN_{NC} model.

5.3.1.5 $\mathbf{MLN_{NC}}$ Modeling

In contrast to the SEMANTIC CORRELATION REASONER which is essentially used to build the MLN_{NC} knowledge base, the following part focuses on using hard axioms extracted from the ontology to enrich our MLN_{NC} model. The considered ontology includes a property REQUIRESUSAGEOFARTIFACT that associates artifacts in the apartment with activities for which they are necessary.

> **Example 5** *Continuing our example, the axiom below defines* PREPHOTMEAL *as a subclass of* PREPAREMEAL *that requires the usage of a cooking instrument:*
>
> $$\text{PrepHotMeal} \sqsubseteq \text{PrepareMeal} \sqcap \exists\text{requiresUsageOfArtifact}.$$
> $$(\text{CookingInstrument} \sqcap (\exists\,\text{isIn}.\{\text{apt}\})).$$

Subsequently, we infer which sensor events must necessarily be observed during the execution of an activity. The following axiom states that "if an event of type *et* is produced by a sensor that detects the usage of an artifact required for executing an activity of class *ac*, then *et* is a *necessary sensor event type* for each activity instance of class *ac*".

$$\text{PRODUCESEVENT}^{-} \circ \text{SENSESUSAGEOF} \circ$$
$$\text{REQUIRESUSAGEOF}^{-} \rightarrow \text{NECESSARYEVENTFOR}.$$

Then, we infer the fillers of the property NECESSARYEVENTFOR through ontological reasoning, translate them in MLN_{NC} axioms, and add them, finally, to the MLN_{NC} model.

Example 6 *Given the introduced axioms, in this case the OWL 2 reasoner infers that* {ET_STOVE} *is a necessary sensor event type for* {AC_PREP_HOT_MEAL}*. Indeed,* ET_STOVE *is produced by usage of* STOVE*, which is the only instance of* COOKINGINSTRUMENT *available in the home.*

Overall, Figure 5.8 depicts our MLN_{NC} model where we distinguish between *observed* (star symbol) and *hidden* predicates. Observed predicates represent knowledge facts, where the instances of hidden predicates are computed by MAP INFERENCE. Semantic correlations are modeled through predicates *PriorProb*, *Event*, and *Instance*. The *PriorProb* predicate represents correlations among sensor events and activities:

$$*PriorProb(SensorEvent, ActivInstance, ActivClass, p)$$

Hence, it describes the probability p that a given sensor event se corresponds to a given activity instance ai of an activity class ac. The probability relies on the semantic correlation between the event type et and the activity class ac (PPM), but also depends on the temporal distance between the sensor event se and the boundaries of the activity instance ai.

Formally, given an activity instance ai of class ac with start time t_{st} and end time t_{ed}, and a sensor event se of type et and timestamp t, the probability p of **PriorProb*(se, ai, ac, p) is computed by the following function:

$$p = \begin{cases} PPM(ac, et) & \text{if } t_{ed}\text{-}MaxDelay_{ac} \leq t \leq t_{st}{+}MaxDelay_{ac} \\ 0 & \text{otherwise} \end{cases}$$

Each sensor event is represented by an instance of the predicate *Event*, which represents the sensor event, its type, and its timestamp:

$$*Event(SensorEvent, EventType, Timestamp)$$

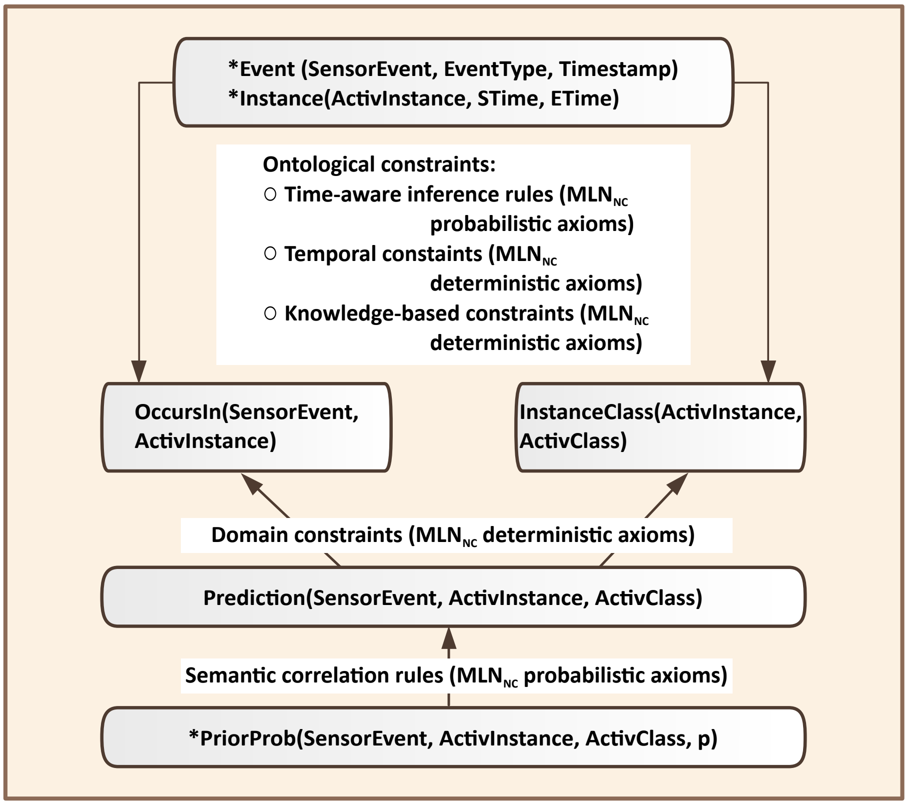


Figure 5.8: Probabilistic activity recognition framework. The arrows indicate the relations and dependencies between the depicted observed and hidden predicates.

Candidate activity instances computed by Algorithm 3 are represented by the predicate *Instance* that models the relation between the activity instance, its start time, and end time:

$$*Instance(ActivInstance, STime, ETime)$$

The instantiated predicates are added as facts to our MLN_{NC} knowledge base and derived from the activity instances and the recorded sensor events.

Hidden predicates and domain constraints. Beside the observed predicates, the model also comprises a set of hidden predicates, which can be considered as our target classes: *Prediction*, *OccursIn*, and *InstanceClass*. The predicate *Prediction* represents the predicted assignment of a sensor event to an activity instance of a given class:

$$Prediction(SensorEvent, ActivInstance, ActivClass)$$

In addition, the other two predicates are used to express domain constraints about the consistency of inferred activity instances:

$$OccursIn(SensorEvent, ActivInstance)$$
$$InstanceClass(ActivInstance, ActivClass)$$

In particular, the following domain constraint states that each sensor event occurs in exactly one activity instance:

$$|ai|OccursIn(se, ai) = 1,$$

While the following one states that each activity instance belongs to exactly one class:

$$|ac|InstanceClass(ai, ac) = 1.$$

Semantic correlation rules. The relations between the observed and hidden predicates are modeled by probabilistic axioms. As illustrated in Figure 5.8, the hidden predicate *Prediction* is derived from *PriorProb*:

$$conf : *PriorProb(se, ai, ac, conf) \Rightarrow Prediction(se, ai, ac).$$

Thus, the confidence value describes the probability that a sensor event is assigned to an activity instance of a given class. In turn, the remaining hidden predicates are derived from the hidden *Prediction* predicate. The corresponding axioms are the following:

$$Prediction(se, ai, ac) \Rightarrow OccursIn(se, ai),$$
$$Prediction(se, ai, ac) \Rightarrow InstanceClass(ai, ac).$$

Note that the above rules are subject to the domain constraints introduced before.

Knowledge-based constraints. Knowledge-based constraints enable us to express conditions about the occurrence (or non-occurrence) of sensor events of a given type during the occurrence of an activity instance.

As mentioned before, knowledge-based constraints are automatically derived from the fillers of the NECESSARYEVENTFOR property obtained from ontological reasoning.

Example 7 *The constraint "each activity instance of type 'preparing hot meal' must be associated to an event of type 'UseStove' " is logically expressed by the rule:*

$$InstanceClass(ai,\ \text{“}PrepHotMeal\text{”}) \Rightarrow \exists\ se, t\ :$$
$$OccursIn(se,\ ai) \wedge *Event(se,\ \text{”}UseStove\text{”},\ t).$$

Temporal constraints. We model MLN_{NC} temporal constraints regarding the duration and the distance of events or activities. We consider two kinds of temporal constraints:

1) *Temporally close events (e.g., whose temporal distance is below* Δ *seconds) likely belong to the same activity instance.* We express this soft constraint through these axioms:

$$\forall\, t_1,\, t_2 \,:\, (|t_1 - t_2| < \Delta) \,\Rightarrow\, tClose(t_1,\, t_2)$$

$$\begin{gathered} w\ Event(se_1,\, et_1,\, t_1) \,\wedge\, Event(se_2,\, et_2,\, t_2) \wedge \\ tClose(t_1,\, t_2) \wedge OccursIn(se_1,\, ai) \,\Rightarrow\, OccursIn(se_2,\, ai) \end{gathered}$$

The latter is a probabilistic axiom whose weight w is chosen experimentally.

2) *Constraints on duration of each activity (e.g. "showering cannot last more than* Δ' *minutes").* We express these constraints either through probabilistic or deterministic axioms, according to the characteristics of the considered activity. Indeed, the variance of the duration of certain activities (e.g. showering) is relatively small, while it is larger for other activities (e.g. preparing dinner). The duration of the former is modeled with deterministic axioms where probabilistic ones are used for the latter. The axioms below state that an instance of "showering" cannot last more than Δ' minutes:

$$\forall\, t_1,\, t_2 \,:\, (|t_1 - t_2| < \Delta') \,\Rightarrow\, tclose\_showering(t_1,\, t_2)$$

$$\begin{gathered} InstanceClass(ai, \text{“}Showering\text{”}) \,\wedge\, OccursIn(se_1, ai) \wedge \\ OccursIn(se_2, ai) \,\wedge\, Event(se_1, et_1, t_1) \wedge \\ Event(se_2, et_2, t_2) \,\Rightarrow\, tclose\_showering(t_1,\, t_2) \end{gathered}$$

Time-aware inference rules. Finally, as explained before, the semantics of some simple activities is naturally expressed in our ontology based on the typical actions composing them. Hence, we apply rules that express the relation of specific operations derived from sensor events in context of time. Consider the following example:

Example 8 *A typical pattern of operations for watering plants consists of (1) "getting water" and (2) "moving to the plants" shortly after. We express this activity inference pattern through the* MLN_{NC} *axioms below:*

$$\begin{aligned} &Event(se_1,\, \text{“}water\_sensor\text{”},\, t_1) \\ &\wedge Event(se_2,\, \text{“}plant\_presence\_sensor\text{”},\, t_2) \,\wedge\, t_1 < t_2 \\ &\wedge tclose\_waterplants(t_1,\, t_2) \,\Rightarrow \exists\, ai \,: \\ &InstanceClass(ai,\, \text{“}WaterPlants\text{”}) \\ &\wedge occursIn(se_1,\, ai) \,\wedge\, occursIn(se_2,\, ai). \end{aligned}$$

5.3.1.6 MAP Inference and Post-processing

In order to reconstruct the relations of activity instances, their class, and the corresponding sensor events, we execute MAP INFERENCE on the presented MLN_{NC} model (see Sections 5.3.1.5) by considering the introduced and generated MLN_{NC} knowledge base (see Sections 5.3.1.4). The result is a set of *OccursIn* and *InstanceClass* predicates. The former maps a sensor event to the most probable corresponding activity instance where the latter assigns the most likely activity class to an activity instance. These (hidden) predicates are POST-PROCESSED in order to detect the class and temporal boundaries of each activity instance ai:

$$AClass(ai) = ac : \exists InstanceClass(ai, ac),$$
$$STime(ai) = \min\{t \,:\, \exists Event(se, et, t) \wedge OccursIn(se, ai)\},$$
$$ETime(ai) = \max\{t \,:\, \exists Event(se, et, t) \wedge OccursIn(se, ai)\}.$$

In this context, $AClass(ai)$ represents the activity class of ai, while $STime(ai)$ and $ETime(ai)$ respectively the start- and end-time. Computing the start and end time of activity instances by the MLN_{NC} resolver would be unnecessarily complicated, hence, they are computed in a post-processing phase. The overall result is a sequence of activities that most likely caused the recorded sensor events.

5.3.2 Online Recognition of Interleaved ADLs

So far, our system only supports offline recognition, i.e., analyzing in batch mode a complete stream of sensor data acquired during a predetermined period. This is sufficient e.g. for a cognitive health assessment of the elderly. Hence, the system monitors the individual's behavior on the long-term and at the end of each day, the system may process all the sensor data acquired during that day. However, this is insufficient in many real-world scenarios. For example, real-time monitoring applications, such as services that require intervention (e.g., reminders, emergency monitoring), require online recognition. Compared to offline recognition, that task is typically harder, since the recognition system must segment the continuous stream of sensor events on the fly in order to infer the most likely activity in nearly real-time and detect activity changes as they happen. For that reason, in the following we present an extension for our introduced system, which enables online activity recognition in a smart-environment.

5.3.2.1 System Overview: Online Recognition Extension

Online recognition has to deal with a continuous stream of sensor events to be processed on the fly. To achieve that we extend our initial approach by two specific layers, namely ONLINE SEGMENTATION and STATISTICAL ANALYSIS OF SEGMENTS (see Figure 5.9). The former layer runs an algorithm that is in charge of inferring a change in the class of the

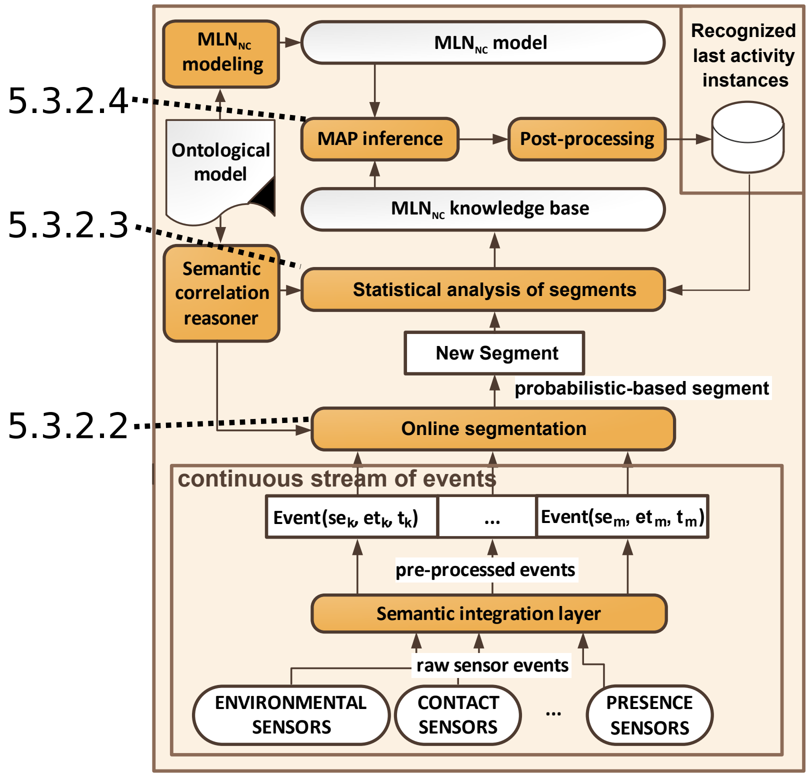


Figure 5.9: Extended system architecture for online recognition. The ONLINE SEGMENTATION module processes the continuous stream of events. The STATISTICAL ANALYSIS layer combines the information received from the sensors and the ontological model to build a *knowledge base*. MAP INFERENCE enables to derive the most probable world from this knowledge base considering the MLN_{NC} *model*. This results in the recognition of the actual activity instances.

current activity performed by the individual, in order to identify possible segments. The latter layer is responsible for identifying activity instance *candidates* derived from the finalized segments. Actually, this layer replaces the STATISTICAL ANALYSIS OF EVENTS layer. The resulting candidates are processed in the same way as in case of offline recognition by the MLN_{NC} reasoner (see Section 5.3.1.6).

In particular, given a temporal sequence of events $\langle ev_1, ev_2, \ldots, ev_n, \ldots \rangle$ where $ev_i = (se_i, et_i, t_i)$, the role of the online segmentation algorithm is to derive a set of segments:

$$\langle Segment(ev_1, \ldots, ev_l), \ldots, Segment(ev_m, \ldots, ev_n), \ldots \rangle,$$

where each segment $Segment(ev_j, ev_{j+1}, \ldots, ev_k)$ represents a set of consecutive and ordered sensor events from ev_j to ev_k. Segments do not overlap and each sensor event is assigned to exactly one segment. The goal of the algorithm is to minimize the number of segments, while ensuring that all the events in a segment are labeled with the same activity class.

Our online segmentation algorithm uses probabilistic and semantic conditions in order to decide whether to finalize a segment and initiate a new one. We call that operation a *split* decision. As soon as a segment is finalized, it is immediately forwarded to the next layer. Subsequently, the STATISTICAL ANALYSIS OF SEGMENTS layer is in charge of connecting the latest finalized segment with the previously generated ones, which in turn also allows to consider previous recognition results. Relations and constraints among activities are taken into account for the re-generation of the MLN_{NC} knowledge base. Periodically, for each new segment, MAP INFERENCE is performed to identify its most probable activity class. In the following, we go into detail and explain these two new layers in detail.

5.3.2.2 Online Segmentation

The ONLINE SEGMENTATION algorithm considers five aspects: *object interaction* (ASP1), *change of context* (ASP2), *consistency likelihood* (ASP3), *time leap* (ASP4), and *change of location* (ASP5). Whenever a new sensor event ev_{new} is detected, all those aspects are evaluated. If at least one aspect determines sufficient conditions to perform segmentation, the current segment is finalized and a new one (with ev_{new} as the first element) is initialized. An advantage of this approach is that the segment length is variable, i.e., it is not necessary to predefine a certain length, which could be usually problematic regarding significant different durations of different activities. In the following, we outline the mentioned aspects:

ASP1) For each object, the system keeps track of its usage status: *in use* or *not in use*. The usage status of each object is automatically updated according to the events in the stream. The *object interaction* aspect finalizes a segment as soon as the system detects that the user stopped interacting with all the objects in the home. For instance, suppose that the type of the current event ev_{new} is “turning off the stove”. If, at the same time, the subject is not actively using any other instrument, the current segment is finalized. Indeed, the current activity is likely terminated. On the other hand, the segment is not finalized if the subject is using other objects at that time (e.g. the oven).

ASP2) The *change of context* aspect considers our ontological model to verify whether the new event in the stream (ev_{new}) is correlated with the last event of the current segment (ev_{last}). In this context, only sensor events related to an interaction are considered, e.g., temperature or presence sensor events are disregarded. Formally, we define

$$possAct(ev(se, et, t)) = \{ac \in \mathbf{A} : PPM(ac, et) > 0\}$$

as the set of possible activities for an event ev given the semantic correlations. If $possAct(ev_{last}) \cap possAct(ev_{new}) = \emptyset$, the aspect derives that ev_{new} cannot be labeled with the same activity class of ev_{last}, and thus the current segment is finalized.

ASP3) The *consistency likelihood* aspect keeps track of the probability that the current segment includes events mostly labeled with the same activity class. Differently from ASP2, in this aspect we consider the whole set of the segment’s events. In particular, we consider the semantic correlation among those events and possible activities, and we finalize the segment if the introduction of the new event ev_{new} determines an abrupt shift in the likelihood of the segment, computed by the following formula:

$$L(S) = \max_{ac_i \in \mathbf{A}} \frac{\sum_{ev_j(se,et,t) \in S} PPM(ac_i, et)}{|S|}, \tag{5.11}$$

where $PPM(ac_i, et)$ is the semantic correlation between activity ac_i and event type et. If the fluctuation of $L(S)$ due to the introduction of ev_{new} in S exceeds an experimentally chosen threshold σ, the current segment is finalized.

ASP4) The *time leap* aspect considers the time distance between consecutive events. If no new event is observed after the most recent event ev_{last} according to a time threshold δ, the current segment is finalized. The value of δ is automatically calibrated based on the stream of sensor events. In particular, we continuously keep track of the third quartile value q of the temporal distances between consecutive sensor events. The value of δ is automatically updated as $2q$ whenever a new segment is finalized. Therefore, the *time leap* aspect is not considered for the very first segment.

ASP5) The *change of location* aspect relies on the fact that most ADLs are performed in a specific location. For that reason, we finalize the segment when the individual moves from a room to a different one. Indeed, there are activities that are performed across several rooms like cleaning but as already mentioned, we aim to have segments with a high purity but in turn we accept if the same activity is divided in several segments.

All these rules are applied simultaneously and continuously, i.e., independently of each other. Further, we do not define a minimal or maximal size of a segment, so, how many events a segment should or has to cover. Finally, when a segment is finalized, it is forwarded to the STATISTICAL ANALYSIS OF SEGMENTS layer that prepares the analysis of this segment and enables to link the new segment with the previously generated segments.

5.3.2.3 Statistical Analysis of Segments

The goal of the STATISTICAL ANALYSIS OF SEGMENTS layer is to generate activity instance candidates based on the finalized segments. Algorithm 4 describes our method in detail. It takes k recent segments and the prior probability matrix as input and it returns activity instance candidates (similar to Algorithm 3). In line 1 and 2, we initialize a *segmentQueue* that includes the segments liable to be merged, as well as the initially empty set of candidates. Then, for each segment s in the queue, we create a new activity instance candidate ai with the same temporal boundaries of s (line 4). We set the class of ai to the most probable activity class according to the PPM (line 5). If that class is the same of another candidate, those candidates are merged (line 7) by extending the end time of the former to the end time of the latter. This operation enables to support interleaved activities. Otherwise, ai is added to instances (line 9). Finally, the set of activity instance candidates is returned (line 12).

Algo 4 Statistical analysis of segments

Input: last k-segments,
prior probability matrix (PPM)
Output: activity instance candidates $\{ai_0, ai_1, \ldots, ai_{j-1}\}$

```
1: segmentQueue ← last k segments
2: candidates ← ∅
3: for each s = Segment(ev_m, ..., ev_n) ∈ segmentQueue do
4:     ai ← new activity instance from time(ev_m) to time(ev_n)
5:     set the class of ai to: argmax_{ac∈A} Σ_{(se_i,et_i,t_i)∈S} PPM(et_i, ac)
6:     if ∃ i ∈ candidates whose class is the same of ai then
7:         replace endtime of i with time(ev_n)
8:     else
9:         candidates ← candidates ⋃ {ai}
10:    end if
11: end for
12: return candidates
```

Hence, in the simplest case, each new segment represents an activity instance candidate. However, considering the last k segments to generate activity instance candidates ensures that the MLN_{NC} resolver has sufficient information to reason the correct activity classes for the corresponding activity instance candidates. In this context, the activity classes that were assigned by the statistical analysis of segments algorithm are only used to merge temporally close candidates. Subsequently these assignments are discarded.

5.3.2.4 MAP Inference and Post-processing

Compared to the basic system so recognizing activities in offline mode, the *MAP Inference* layer is still the same (cf. see Section 5.3.1.6). Hence, the underlying MLN_{NC} model is unchanged and is only generated once but the corresponding MLN_{NC} knowledge base is recreated as soon as a new segment was finalized. Consequently, the MLN_{NC} resolver is also executed several times where the result is post-processed in respect of merging new activity instances with previous generated ones of the same activity class. Subsequently, the gained knowledge can be used to enrich the subsequent generated knowledge bases, e.g., adding which activities were already recognized.

5.3.3 Active Learning in a Smart-Environment

In order to cope with the incompleteness of an ontology and the heterogeneity of environments and individuals, we also introduce a collaborative active learning process to refine the correlations derived by the ontology. The stream of sensor events is segmented in real-time, and based on the discrimination value of correlations on the segment, a feedback may be asked to the subject about which activity is being performed. Feedback responses coming from different homes are collected in a cloud infrastructure and each home receives personalized information to refine its recognition model. The collaborative active learning feature of our system (see Figure 5.10) also deals with the common situation in which a new device is installed in the infrastructure, by producing a new set of correlation values regarding the new device events.

5.3.3.1 System Overview: Active Learning Extension

Commonly, the ontological model is necessarily limited to specific environments and activities as it was manually designed by knowledge engineers with a specific application in mind. Thus, our *semantic correlations* may not be sufficiently comprehensive to cover different application domains. Moreover, some sensor event types (e.g., motion or ambient sensors) do not convey any explicit semantic information; hence, no semantic correlation can be inferred for these event types from the ontology. For this reason, our system collects *feedback items* from the smart-homes in order to discover semantic correlations not inferred from the ontology. For acquiring a feedback, the system interactively queries the user to provide the class of the current ADL. Acquired feedback is collaboratively shared among the smart-homes to update semantic correlation values in a personalized fashion. For clarification, in the following we name *origin* the environment (home and resident) providing feedback, and *target* the environment where feedback is used to update semantic correlations.

The feedback acquisition mechanism relies on our concept of *segments*. As soon as the system determines that a segment's events do not provide enough hints to determine reliably its activity class according to an information-theoretic metric, it queries the user to obtain a feedback. For this purpose, the ONLINE RULE-BASED SEGMENTATION layer (see Figure 5.10) is in charge of segmenting the continuous stream of sensor events. The segmentation method is based on the introduced semantic rules, i.e. objects interaction (ASP1), time constraints (ASP4), and change of location (ASP5). The role of these rules is to group together those consecutive events that most likely originate from the same activity instance. As soon as a segment is finalized, it is processed by the QUERY DECISION layer in order to decide whether triggering a feedback query or not. That module processes the segment to apply an information-theoretic metric considering the segment's events and the semantic correlations. If the activity class is uncertain according to that metric, the module triggers a feedback query. A user-friendly and unobtrusive interface is in charge of issuing the feedback query and collecting the answer of the resident.

The acquired *feedback* is transmitted to a *Cloud Service*, where the COLLABORATIVE FEEDBACK AGGREGATION layer is in charge of computing personalized feedback items for the different environments. Personalization is based on the similarity between the origin and target environment. The *Cloud Service* periodically sends personalized feedback items to each target. Received feedback is used by the SEMANTIC CORRELATIONS UPDATER layer to discover novel semantic correlations and to update the values of existing ones.

For the sake of this work, we assume that the *Cloud Service* is trusted. However, in a real deployment it would likely be an *honest-but-curious* third party. Proper privacy

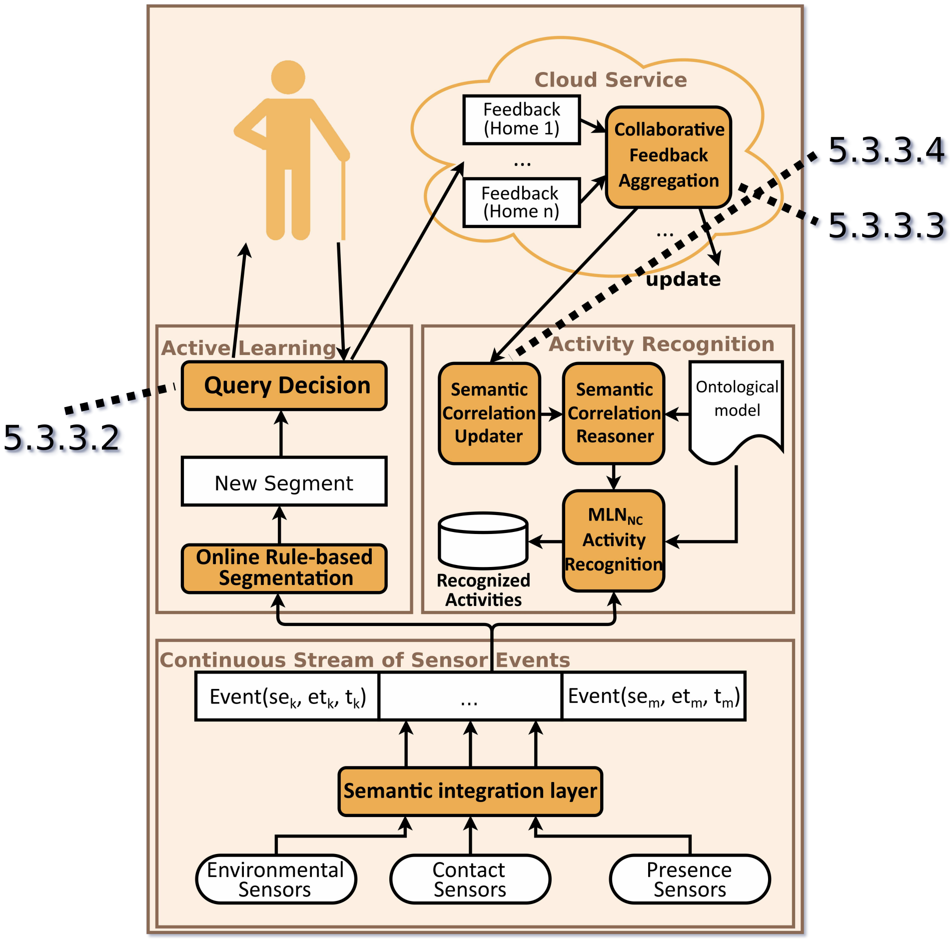


Figure 5.10: Extended system architecture for active learning in a smart-environment. The ONLINE RULE-BASED SEGMENTATION layer uses the previous introduced aspects (see Section 5.3.2.2) to identify suitable segments for deciding to query the user (QUERY DECISION). In case of querying, the result is forwarded to the *Cloud Service* for COLLABORATIVE FEEDBACK AGGREGATION of the different homes. At a certain point in time, the processed feedback is forwarded to the individual homes to update the semantic correlations (SEMANTIC CORRELATION UPDATE). The remaining components are unchanged (cf. see Figure 5.9).

techniques are thus needed to protect sensitive data and at the same time to preserve the CLOUD SERVICE functionalities. We will come back to this issue when we start the discussion (see Section 5.5.3).

5.3.3.2 Query Decision

Given a segment S, the QUERY DECISION layer decides if it is necessary to query the resident. In particular, if the *semantic correlations* of the event types in S are inconclusive when considered together (i.e., they do not converge on a specific activity class), we ask the resident which activity was actually performed. For that purpose, we introduce the concept of a *segment's bag*:

$$Bag(S) = \{et \mid ev = (se, et, t) \in S\}$$

where S is a finalized segment and $Bag(S)$ is a bag (i.e., a multiset) which contains the types of the events contained in S. It is important to note that the temporal order of events of a segment is not reflected by its bag. Hence, for each bag $Bag(S_i)$, we compute for all $ac \in \mathbf{A}$ the likelihood that the segment S_i represents an activity instance of ac. This is computed as follows:

$$L(ac \mid S) = \frac{\sum_{et \in Bag(S)} PPM(ac, et)}{|Bag(S)|}$$

where $PPM(ac, et)$ is still the *semantic correlation* between ac and et (see Section 5.3.1.3).

After we compute $L(ac|S)$ for all activity classes, we normalize these values in order to have a probability distribution. Subsequently, the *entropy* is calculated on the distribution to determine the system's confidence for the segment S:

$$H(S) = \sum_{ac \in A} P(X = ac \mid S) \cdot log(\frac{1}{P(X = ac \mid S)})$$

where $P(X = ac \mid S)$ results from the normalized $L(ac \mid S)$ values.

Finally, if $H(S)$ is higher than a predefined threshold λ, the system ranks S as *uncertain*. In this case, the system queries the resident in order to provide an activity label ac for S, and each event type $et \in Bag(S)$ is associated with ac. These associations are transmitted immediately to the *Cloud Service* together with the identification of the origin.

Note that segments containing noisy events which occurred outside activities execution (e.g., trigger of presence sensors) would likely lead to high entropy values. To overcome this issue, we rely on the SEMANTIC INTEGRATION LAYER (presented in Section 5.3.1.1) to reduce as much as possible the generation of those noisy events. Moreover, in order to reduce further noisy data, we also discard segments with few events.

In the following, we describe our collaborative adaptation framework, which relies on two main components. The COLLABORATIVE FEEDBACK AGGREGATION layer (which runs on the *Cloud Service*) collects and aggregates the *feedback* received from the several homes and it periodically transmits personalized updates to each target home. On the other hand, the SEMANTIC CORRELATION UPDATER algorithm (which runs in the home's gateway) is in charge of analyzing the personalized update in order to improve the semantic correlations produced by the ontology.

5.3.3.3 Collaborative Feedback Aggregation

The *Cloud Service* continuously receives and stores feedback transmitted by the participating homes. Each feedback item f is represented by a vector

$$f = \langle et, ac, o \rangle$$

where et is an event type, ac is an activity class, and o is the origin of the feedback.

Based on the received feedback, the *Cloud Service* periodically transmits *personalized feedback items* to each target home. A personalized feedback item is represented by a vector $\langle et, ac, p, s\rangle$, where $p \in (0, 1]$ is the *predictiveness* of event type et for activity class ac computed based on feedback items, and $s \in (0, 1]$ is the *estimated similarity* between the feedback origins and target. More precisely, the similarity s is computed based on the similarity between the smart-home infrastructures (sensor networks) but also considers the similarity between the respective residents. The idea is to consider the similarity to weight the personalized feedback.

The COLLABORATIVE FEEDBACK AGGREGATION layer is in charge of computing personalized feedback items based on the received feedback. In order to measure the similarity between the origin and target of a feedback, that module relies on a similarity function $sim : H \times O \to [0, 1]$, where H is the set of targets, and O is the set of origin environments. The output of $sim(h, o)$ is a value between zero and one. Of course, the most appropriate definition of the target environment features, as well as the method to compute sim values, depend on the addressed application.

Based on a multiset F of feedback items, the module computes personalized feedback items for each target environment. In particular, consider a target h. At first, for each event type et and activity class ac, the following formula computes the personalized feedback *support*:

$$supp(et, ac, h, F) = \sum_{f=\langle et,ac,o\rangle \in F} sim(h, o).$$

In order to exclude unreliable feedback, the *Cloud Service* transmits only personalized feedback whose support is larger than a threshold σ. For each reliable personalized feedback, the module computes its *predictiveness* value:

$$pred(et, ac, h, F) = \frac{supp(et, ac, h, F)}{\sum\limits_{ac_i \in \mathbf{A}} supp(et, ac_i, h, F)},$$

This is the normalization of et's support values, distributed over all the activity classes.

Finally, the module computes the *estimated similarity* as the median value of the similarity between the feedback items' origin and the target:

$$s(et, ac, h, F) = \underset{f=\langle et,ac,o\rangle \in F}{median} sim(h, o).$$

5.3.3.4 Semantic Correlation Updater

Periodically, each home receives an update from the CLOUD SERVICE consisting of a set $\mathbf{P}$ of *personalized feedback items*. The SEMANTIC CORRELATION UPDATER algorithm analyzes $\mathbf{P}$ along with the semantic correlations inferred by the ontology in order to refine the semantic correlations. In the following, we denote $SC(et, ac)$ as the semantic correlation between et and ac computed by our algorithm.

Algo 5 Semantic Correlation Updater

Input: set of personalized feedback items $P = \{\langle et_1, ac_1, p_1, s_1\rangle, \langle et_2, ac_2, p_2, s_2\rangle, \dots\}$, semantic correlation function PPM computed by the ontology, and set U of unpredictive events

Output: refined semantic correlation function SC

```
SC ← PPM
newevents ← ∅
for each ⟨et, ac, c, s⟩ ∈ P do
    if et ∈ U then
        SC(et, ac) ← c
        if et ∉ newevents then
            newevents ← newevents ∪ {et}
            for each ac_i ∈ A s.t. ac_i ≠ ac do
                SC(et, ac_i) ← 0
            end for
        end if
    else if PPM(ac, et) = 0 then
        ac_ont ← an activity ac_j ∈ A s.t. PPM(et, ac_j) > 0
        SC(et, ac_ont) ← SC(et, ac_ont) / (1 + s · SC(et, ac_ont))
        SC(et, ac) ← s · SC(et, ac_ont)
        for each ac_i ∈ A do
            if ac_i ≠ a_ont and ac_i ≠ ac then
                SC(et, ac_i) ← SC(et, ac_i) · (1 − SC(et, ac))
            end if
        end for
    end if
end for
return SC
```

The pseudo-code of the SEMANTIC CORRELATION UPDATER algorithm is shown in Algorithm 5. At first, the algorithm initializes the current semantic correlations with the ones computed by the ontology (PPM). Then it initializes the set U of *unpredictive event types*:

$$predAct(et) = \{ac \,|\, et \text{ is a predictive event for } ac\}$$

$$U = \{et \,|\, predAct(et) = \emptyset\}$$

U contains all the event types which the current ontology does not consider predictive for any activity. Then, the algorithm iterates on each *personalized feedback item* $\langle et, ac, p, s\rangle$ contained in $\mathbf{P}$ in order to update the semantic correlations produced by the ontology. If et belongs to U, $SC(et, ac)$ is set to its predictiveness value p. Moreover, if et is observed for the first time during the current iteration (i.e., if it is not yet part of the set *newevents*), the semantic correlation value $SC(et, ac_i)$ for any other activity class $ac_i \neq ac$ is initialized to 0, and et is added to the set of new events. Intuitively, since *unpredictive event types* have uniform semantic correlations for all the activities, they are usually queried more than other event types since they contribute most in increasing the entropy value. This makes the *predictiveness* values provided by the *Cloud Service* reliable to be used as semantic correlations for et, thus overriding the uniform semantic correlations inferred by the ontology.

In the case of $et \notin U$, we update the semantic correlations only if $SC(et, ac) = 0$. Indeed, our algorithm does not modify the non-zero semantic correlations inferred by the ontology, since they are considered reliable. Instead, whenever a new semantic correlation between et and ac is discovered from a *personalized feedback item*, it is necessary to correspondingly scale all the other semantic correlations regarding et so that $SC(et, ac)$ remains a distribution probability (i.e., $\sum_{ac \in \mathbf{A}} SC(et, ac) = 1$).

Hence, we select a random activity ac_{ont} correlated to et according to the ontology (i.e., such that $PPM(et, ac_{ont}) > 0$). Then we scale $SC(et, ac_{ont})$ considering the *estimated similarity* value s:

$$SC(et, ac_{ont}) := \frac{SC(et, ac_{ont})}{1 + s \cdot SC(et, ac_{ont})}$$

Since the event types for which the ontology already provided a semantic correlation are generally less queried, it is not reliable to use the predictiveness value to update the semantic correlations. This is why we use the *estimated similarity* s instead. The next step consists in updating $SC(et, ac)$:

$$SC(ac, et) := s \cdot SC(et, ac_{ont})$$

Finally, we update the semantic correlations of all the remaining activities ac_j (such that $ac_j \neq ac_{ont}$ and $ac_j \neq ac$) in the following way:

$$SC(et, ac_j) := SC(et, ac_j) \cdot (1 - SC(et, ac)).$$

It can be easily verified that, by construction, Algorithm 5 enforces that given an event type et, the revised $SC(et, ac)$ function is a probability distribution over all ac values. After each update, the function $SC(et, ac)$ computed by our algorithm thus replaces $PPM(ac, et)$ for both the QUERY DECISION and MLN_{NC} ACTIVITY RECOGNITION layers.

5.4 Experimental Results

In the following, we present our experimental setup and results. The presentation order is consistent compared to the introduced methods and the results are compared across the introduced approaches for discussion. The corresponding MLN_{NC} model and the ontology are available[3]. Unless otherwise specified, the presented results rely on the introduced unsupervised approach, where the semantic correlations (PPM matrix) were derived by ontological reasoning. For the evaluation, we use the introduced datasets *CASAS* [17,186] and *SmartFABER* [188] (see Section 5.1). Both datasets include interleaved activities in a smart-home environment. As before, *F-measure* is considered as synonym of F_1*-measure*. To provide the possibility to reconstruct our approaches and experiments, we point to a REST API and web interface, which provides the considered MLN_{NC} solver[4]. In particular, we focus on the following research questions:

[3] `https://sensor.informatik.uni-mannheim.de/#results2016unsupervised`
[4] `http://executor.informatik.uni-mannheim.de`

RQ2.1 Which method can be used to overcome the requirement of a large expensive labeled dataset of Activities of Daily Living?

RQ2.2 Which type of recognition method is suitable for handling the diversity and complexity of Activities of Daily Living?

RQ2.3 How can external sensor events be exploited to recognize Activities of Daily Living in almost real-time?

RQ2.4 Given a generic model of a smart environment, how can it be adapted to a certain environment and user at run-time?

The following subsections belong to the publications [2, 5, 8].

5.4.1 Recognizing Interleaved Activities of Daily Living

To evaluate the effectiveness of semantic correlations extracted with our method, we also performed experiments computing the PPM from the dataset; more precisely, based on the frequency of the sensors types produced by the different activities. We denote by MLN_{NC} (Ontology) the former method and by MLN_{NC} (Dataset) the latter.

5.4.1.1 CASAS Dataset

During this experiment, we evaluated how well the considered sensor events could be assigned to the corresponding activity instance, but also the quality of detected activity boundaries. Knowing the start and end time of a performed activity allows to assign filtered or noisy sensor events afterwards (e.g., movement). In this context, we analyze each patient separately and focus on all sensor events at once (i.e. the stream is not segmented). Considering our model, we created general and transferable rules and do not rely on any kind of movement patterns or specific behavior that only occur in this scenario. Hence, we focused on the interaction with objects and their dependencies as well as the introduced temporal constraints that should prevent misinterpretation.

Table 5.4 shows that our method outperforms the HMM approach used in [186] in assigning each sensor event to the activity instance that generated it. We observe that we recognize each ADL at least equal or better than HMM, except *Clean*. The poor performance in recognizing *Clean* is because, in the CASAS dataset, it is characterized by different movement patterns that are only partially captured by our method, especially when semantic correlations are extracted from the ontology. Considering the other ADLs, the PPM generated by ontological reasoning obtains essentially the same performance of the one extracted from the dataset, confirming the effectiveness of our semantic correlation reasoner.

Focusing on the other ADLs, the experiments show that the interactions with objects are strong indicators of the performed activities. However, inspecting the recognition result in detail, we noticed a few cases in which subjects exhibited strange behaviors;

e.g., prepared soup without water or took the phone but did not place a phone call. Especially the latter case is hard to recognize without further information. The former case is probably related to sensor errors.

Table 5.4: CASAS dataset: Results (F-measure) of the proposed activity recognition method compared to related work for interleaved activities. *Dataset* (supervised) and *Ontology* (unsupervised) describe the source of semantic correlations (PPM matrix).

Class	**HMM** [186] **(time-shifted)**	$\mathbf{MLN_{NC}}$ **(Dataset)**	$\mathbf{MLN_{NC}}$ **(Ontology)**
ac_1	0.66	0.80	**0.85**
ac_2	0.86	**0.88**	0.81
ac_3	0.29	**0.74**	0.72
ac_4	0.60	0.69	**0.72**
ac_5	**0.83**	0.81	0.81
ac_6	0.83	0.87	**0.88**
ac_7	**0.88**	0.78	0.57
ac_8	0.67	**0.90**	0.88
avg.	0.70	**0.81**	0.78

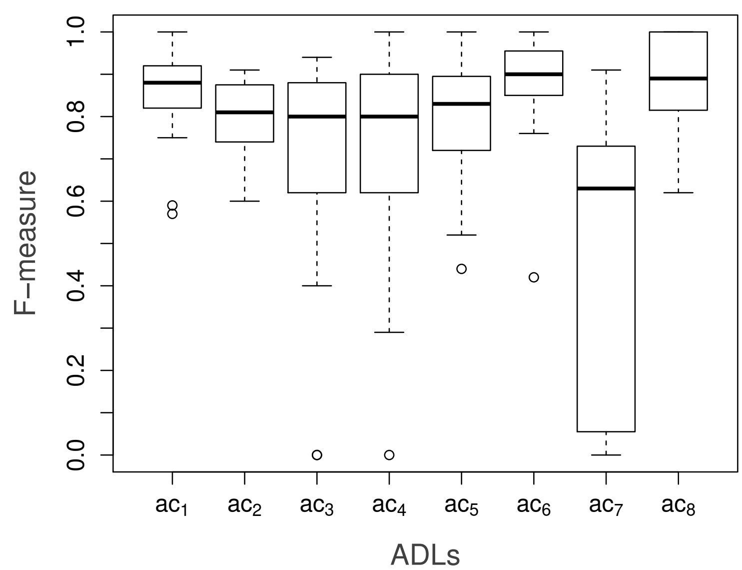


Figure 5.11: CASAS dataset: Detailed recognition results for each ADL, aggregated over all subjects and represented by a box plot. Circles indicate outliers and the box represent the lower and upper quartile.

Figure 5.11 illustrates the individual results in more detail. It highlights that there are cases where we could not recognize the activities *Answer the phone* and *Clean* at all, but in general the distribution is very similar and narrowed.

Considering the boundary detection method, the experiments show that preceding results and the quality of the detected boundaries for the individual activities are weakly related. Table 5.5 describes the deviation from the actual boundaries in detail. ΔStart is the average difference between the actual and predicted start of an activity instance in minutes. ΔDur is the average difference of actual and predicted duration. In context of the typical duration of each ADL, the boundaries are well detected. Hence, the highest deviations are associated with the longest ADLs, and the overall results are acceptable for most applications.

Table 5.5: CASAS dataset: Results of boundary detection with MLN_{NC} (Ontology). It shows the average deviation [min] of the candidate compared to the refined instances.

Class	ΔStart (Candidate)	ΔStart (Refined)	ΔDur (Candidate)	ΔDur (Refined)
ac_1	**0.67**	0.77	1.44	**0.89**
ac_2	**0.59**	**0.59**	**2.97**	3.14
ac_3	**0.07**	0.08	0.93	**0.83**
ac_4	**0.08**	**0.08**	**0.34**	0.42
ac_5	1.30	**1.08**	5.81	**4.64**
ac_6	1.62	**0.11**	4.08	**0.80**
ac_7	1.31	**0.70**	2.39	**2.25**
ac_8	**0.08**	0.10	1.30	**0.52**
avg.	0.73	**0.46**	2.42	**1.70**

When we compare the candidate instances and the refined results obtained through MLN_{NC} reasoning, it strikes that our method refines the candidates reliably. Regarding *watch DVD* (ac_2) and *answer the phone* (ac_4), the refined duration increased slightly, because in some cases subjects took the phone well before using it, or turned on the DVD player well before watching a DVD. Besides, the low numbers clearly show that the duration of the different ADLs was in general short.

5.4.1.2 SmartFABER Dataset

In order to be comparable with the results of previous works on the same dataset, we focused on activity instance classification. Table 5.6 shows the corresponding results and indicates that the accuracy achieved by our unsupervised method is comparable to the one achieved by the supervised method used in [14]. That method relied on temporal-based feature extraction and on a Random Forest classifier. However, we were unable to recognize *eating* because in the dataset it was only characterized by a single presence sensor close to the table, which was also triggered in context of the other activities. The results (see Table 5.6) may indicate that the recognition rate is acceptable but the boundary results clarify that the corresponding activity instance is stretched significantly beyond the actual activity instance, as the system cannot distinguish when the presence sensor is triggered by eating or a non-eating activity. Besides, we were able to recognize *others*, which was not considered in [14].

Inspecting the results, we notice that with *cooking* our unsupervised method achieves essentially the same recognition rate of the supervised technique. With *taking medicines*, the accuracy of our method is lower, mainly due to the absence of sensors strongly correlated to that ADL. The accuracy of recognizing *others* is in line with the one of the other activities. Considering the corresponding instance boundary results, Table 5.7 shows that, also with this dataset, MLN_{NC} refinement significantly improves the accuracy of predicted activity instances. However, we have higher delta values with respect of the previous experiment. This is because activity instances of this dataset have a much longer duration with respect of the ones in CASAS dataset. The obtained results indicate a correlation between instance recognition results and the quality of boundary detection. For instance, *taking medicines*, which showed best instance recognition results, is related with the smallest error on boundaries. The boundary error of *cooking* is higher but still reasonable, since this activity can potentially last more than one hour. The worst results are obtained, also in this case, with *eating*, mainly due to the above-mentioned problems: the boundary error is so large because other activity instances, which happened in the same location, produced many *eating* false positives, hence extending the boundaries of the instances of this class. In general, considering the small set of activities, we state there is an evidence that our current approach is reliable if there is at least some kind of semantic relation between sensor events.

5.4.2 Online Recognition of Interleaved ADLs

Compared to the preceding experiments, the additional challenge introduced by online recognition consists in the need for segmenting the continuous stream of sensor events on the fly. For that reason, this time we investigate not only activity recognition quality, but also quality of segments. In the following, first we propose two metrics to evaluate the overall segmentation quality: *purity* and *deviation of segments* (DS for brevity). Subsequently, we present the corresponding experimental results and the overall performance of the recognition system.

Table 5.6: SmartFABER dataset: Results (F-measure) of the proposed activity recognition method compared to related work. *Dataset* (supervised) and *Ontology* (unsupervised) describe the source of semantic correlations (PPM matrix).

Class	**SmartFABER** [14] **(supervised)**	$\mathbf{MLN_{NC}}$ **(Dataset)**	$\mathbf{MLN_{NC}}$ **(Ontology)**
ac_9	**0.95**	0.84	0.83
ac_{10}	**0.76**	0.67	0.75
ac_{12}	-	0.67	**0.70**

Table 5.7: SmartFABER dataset: Results of the boundary detection method. It shows the average deviation [min] of the candidates compared to the refined instances.

Class	**ΔStart (Candidate)**	**ΔStart (Refined)**	**ΔDur (Candidate)**	**ΔDur (Refined)**
ac_9	**2.20**	2.53	**1.08**	**1.08**
ac_{10}	14.44	**8.95**	25.83	**21.13**
ac_{12}	7.56	**3.26**	34.17	**16.59**
avg.	8.07	**4.91**	20.36	**12.94**

5.4.2.1 Segmentation Evaluation Metrics

A segment S is perfectly pure (i.e., its purity value is equal to 1) when all of its events $ev_i \in S$ are labeled with the same activity class. The formula to compute the purity of a segment S is given below:

$$purity(S) = \max_{ac \in \mathbf{A}} \sum_{ev_i \in S} \frac{\mathbb{1}[ev_i \text{ is labeled } ac]}{|S|} \tag{5.12}$$

Because we aim at generating segments covering a single activity instance, our goal is to obtain segments as pure as possible. Since our segmentation algorithm produces segments with dynamic size, we compute the overall purity of a set of segments $\mathbf{S}$ as the average of $purity(S)$ $\forall S \in \mathbf{S}$, weighted according to the size of each segment:

$$overallPurity(\mathbf{S}) = \frac{\sum\limits_{S \in \mathbf{S}} purity(S) \cdot |S|}{\sum\limits_{S \in \mathbf{S}} |S|} \tag{5.13}$$

However, purity alone is not sufficient to measure the effectiveness of a segmentation algorithm. For instance, an algorithm instantiating a new segment for each sensor event would achieve maximum purity, but would be of little utility, since inferred segments would not resemble the exact ones. Indeed, an exact segmentation algorithm initiates a new segment only when consecutive events belong to different activity classes. For this reason, we also compute DS, as the root mean square of the segmentation error in terms of the number of inferred segments. Formally, considering a sequence of sensor events $E = \langle\, ev(se_1, et_1, t_1), \ldots, ev(se_n, et_n, t_n)\,\rangle$, we denote $S_{E,A}$ the set of segments for E predicted by a segmentation algorithm A, and we denote $\overline{S}_E$ the exact set of segments of E. The segmentation error $\epsilon(S_{E,A}, \overline{S}_E)$ is computed as the modulus of $|S_{E,A}| - |\overline{S}_E|$. Hence, given a set of sequences of sensor events $\mathcal{E} = \{E_1, E_2, \ldots, E_j\}$, we compute the DS of A by the following formula:

$$DS(\mathcal{E}, A) = \sqrt{\sum_{E \,\in\, \mathcal{E}} \frac{\epsilon(S_{E,A}, \overline{S}_E)}{|\mathcal{E}|}}$$

5.4.2.2 CASAS Dataset

In order to show the effectiveness of our segmentation technique, we compare our method with a simpler one (which we call *Naive Segmentation*) which performs segmentation by using a static sliding window that covers w sensor events and has a window overlap factor o. We have empirically determined that the best parameters for this dataset are $w = 6$ and $o = 50\%$. In addition, we also perform experiments considering different combinations of the introduced aspects ASP1, ASP2 and ASP3 (see Section 5.2.2), since they had the highest impact on segmentation's quality. We report only those combinations which

Table 5.8: CASAS dataset: Recognition performance ($\mathbf{F_1}$ measure) of the basic system (Offline Mode, cf. Section 5.4.1.1) and the online extension (Online Mode) compared with a naive segmentation approach and a supervised method based on Hidden Markov Model (HMM).

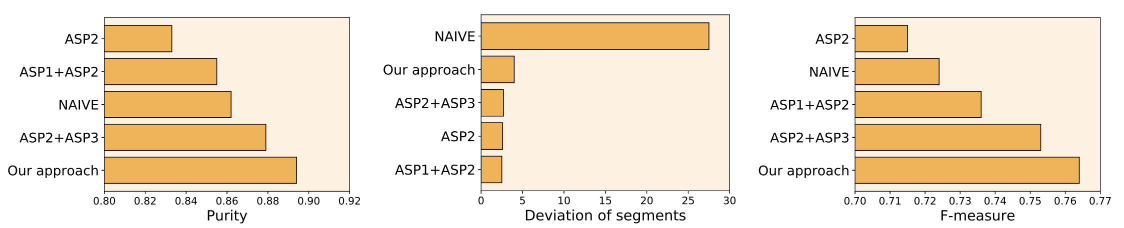
Class	**HMM** [186] (cf. Table 5.4)	**Offline Mode** (cf. Table 5.4)	**Naive Segmentation**	**Online Mode**
ac_1	0.66	**0.85**	0.71	0.74
ac_2	**0.86**	0.81	0.78	**0.86**
ac_3	0.29	**0.72**	0.44	0.62
ac_4	0.59	0.72	**0.74**	**0.74**
ac_5	0.83	0.81	0.89	**0.93**
ac_6	0.83	**0.88**	0.82	**0.88**
ac_7	**0.88**	0.57	0.70	0.56
ac_8	0.67	**0.88**	0.67	0.77
avg.	0.70	**0.78**	0.72	0.76

reached satisfactory recognition results. For instance, we notice that aspect ASP4 (*time leap*) has no impact on this dataset, since sensor events are temporally close together.

Figure 5.12 shows how *purity*, DS, and overall F-measure change by varying the segmentation algorithm. Even if our system (i.e., where we use all the five aspects) does not achieve the lowest DS value, it achieves the best *purity* and the best recognition results with respect to the considered segmentation techniques. Analyzing the results of the naive approach, it emerges that it reaches an acceptable *purity*, but it is affected by a high DS value. This is because the naive segmentation technique produces a high number of segments, negatively influencing recognition results.

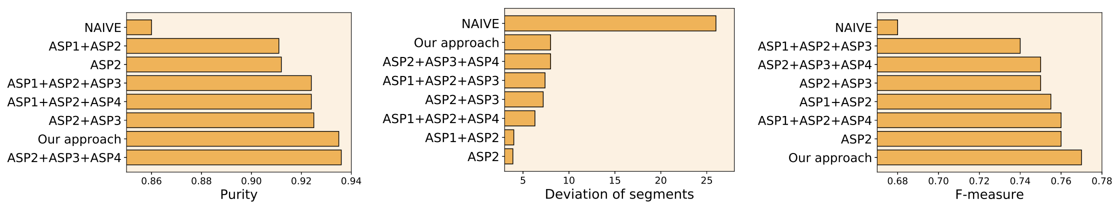
Table 5.8 shows that our method still outperforms the HMM approach [186] in assigning each sensor event to the activity instance that generated it. Moreover, using our segmentation strategy results also in a better performance than using the naive approach (+4%). In this context, the F-measure of the individual ADLs is always comparable (±1%) or higher (up to 17%). Comparing the offline and online modes, the recognition results are similar except for *fill medication dispenser* (ac_1), *water plants* (ac_3) and *choose outfit* (ac_8).

In case of *fill medication dispenser* (ac_1) and *water plants* (ac_3), these activities are essentially recognized by specific events that have to be temporally close. For instance,



(a) Segments average *purity* **(b)** Deviation of segments (DS) **(c)** ADLs recognition's F-measure

Figure 5.12: CASAS dataset: How *purity*, deviation of segments (DS) and F-measure vary by changing the online segmentation technique.



(a) Segments average *purity* **(b)** Deviation of segments (DS) **(c)** ADLs recognition's F-measure

Figure 5.13: SmartFABER dataset: How *purity*, deviation of segments (DS) and F_1 vary by changing the online segmentation technique.

water plants is characterized by the events *"opening the kitchen cupboard"* and *"taking water"*. Unfortunately, our segmentation technique often separates those events in different segments as they are not exclusively related to a single activity and subjects usually performed other interleaved activities. Regarding *choose outfit* (ac_8), looking closely at the data, we noticed that usually this activity has a long duration and most related sensor events are also related to other activities. These facts trigger ASP3 (consistency likelihood) to initiate unnecessary segments, negatively influencing recognition rates.

On the other side, the activities *watch DVD* (ac_2) and *prepare birthday card* (ac_5) are significantly better recognized by the online algorithm. Indeed, those activities can be better recognized when isolated in specific segments and separated from possibly noisy sensor events belonging to other activities.

Considering the overall results, we claim that the decrease of accuracy (at most -2%) introduced by online segmentation is sufficiently small to preserve the utility of predictions for most applications.

5.4.2.3 SmartFABER dataset

As in the preceding presented results, Figure 5.13 shows the segmentation's quality of our method compared to the naive approach and to selected combinations of the introduced aspects (see Section 5.2.2). With respect to those combinations, our approach (i.e. using all aspects) achieves high *purity*. The combination *ASP2+ASP3+ASP4* achieves slightly higher purity; however, the difference is negligible. Further, the DS of our system is also significantly smaller than the one of the naive approach. Overall, we obtain the best recognition results in terms of overall F-measure.

Concerning the naive approach, the results are clear, i.e., the *purity* is the lowest and the DS is the highest. Thus, results obtained by naive segmentation with this dataset are even worse with respect to the ones achieved with the CASAS dataset. As in the preceding experiments, we have determined the optimal parameters empirically: $w = 4$ and $o = 50\%$. According to our understanding, this fact indicates that high variability of activity execution (motivated by cognitive impairment of the subject in this dataset) calls for sophisticated segmentation strategies. Summarizing, our system recognized ADLs 9%

better than the naive approach and improves the considered combinations of aspects up to 3%.

Table 5.9 shows detailed recognition results and indicates that the accuracy achieved by our unsupervised method is comparable to the one achieved by the supervised method used in [14]. That method relied on temporal-based feature extraction and on a Random Forest classifier. Further, we are still unable to recognize *eating* (ac_{10}) (cf. Offline Mode, see Section 5.4.1.2) due to the mentioned reasons.

The results also show that the online mode performs very similar to its offline counterpart on this dataset.

Table 5.9: SmartFABER dataset: Recognition performance (F-measure) of the basic system (Offline Mode, cf. Section 5.4.1.2) and the online extension (Online Mode) compared with a naive segmentation approach and a supervised method (SmartFABER).

Class	**SmartFABER** [14] (cf. Table 5.6)	**Offline Mode** (cf. Table 5.6)	**Naive Segmentation**	**Online Mode**
ac_9	**0.95**	0.83	0.74	0.81
ac_{10}	**0.76**	0.75	0.65	**0.76**
ac_{12}	-	0.70	0.67	**0.71**
avg.	-	0.76	0.69	0.76

5.4.3 Active Learning in a Smart-Environment

In following experiments, we evaluate our active learning component, i.e., how personalized feedback items received from similar homes/subjects can affect activity recognition rates. For this purpose, we only use the CASAS dataset as the SmartFABER dataset covers only a single person. Hence, we simulate 21 apartments with identical sensing infrastructures but inhabited by different subjects. This setup resembles the one of a residence for elderly people consisting of several similar apartments. We fixed the similarity $sim(h_1, h_2)$ between each pair of apartments to 0.5, since the sensing infrastructures are identical (i.e., their similarity is 1), while the profiling of the subjects is unknown.

During a pre-processing phase, we excluded motion sensors that we found out to be *noisy*; i.e., producing measurements essentially independent from the performed activities. Most of these noisy motion sensors were placed in locations irrelevant for the activity recognition task. Other ones triggered too many events, possibly due to excessively high sensitivity or too wide coverage area. Hence, we kept motion sensor events from seven devices only[5].

We performed *leave-one-subject-out* cross validation. In each fold, the system collects feedback items from 20 subjects and uses them to update semantic correlations for the remaining one.

Table 5.10 and Figure 5.14 summarizes our overall experimental results. The results show that the application of our collaborative active learning method increases recog-

[5]Those sensors are identified as M02, M03, M04, M05, M13, M23, and M24 in the dataset.

Table 5.10: Results (F-measure) of the proposed ADL recognition method compared to related work for interleaved activities.

Class	Machine Learning (supervised) [192]	Probabilistic Logic (unsupervised) [53]	Our Approach (w/o active learning)	Our Approach (w/ active learning)
ac_1	0.80	0.74	0.78	**0.82**
ac_2	**0.87**	0.84	0.85	**0.87**
ac_3	0.59	0.36	0.70	**0.71**
ac_4	0.52	0.49	0.67	**0.72**
ac_5	**0.88**	0.83	0.77	0.78
ac_6	0.85	0.67	**0.89**	**0.89**
ac_7	0.57	0.36	0.46	**0.63**
ac_8	**0.84**	0.69	0.71	0.82
avg.	0.74	0.70	0.73	**0.78**

nition performance of about 5%. In order to compare our system with state-of-the-art techniques, we also implemented the supervised method proposed in [192], which relies on machine learning and time-based feature extraction. As machine learning algorithm, we used Random Forest, since it is commonly used in activity recognition systems and it already performed very well in our previous experiments (see Section 4.4). We executed the experiments using that method with the same dataset using leave-one-subject-out cross validation. Results show that our system outperforms the supervised method in terms of average F-measure, and achieves equal or better results in recognizing 6 out of 8 ADLs. The supervised technique performs significantly better in recognizing *prepare birthday card* (ac_5). The main reason is that the classifier was trained on temporal-based features that represent relations between sensor events. Thus, the order of certain sensor events but also their temporal distance leads to a reliable pattern for ac_5 in this dataset. We also compared the system with a recent unsupervised method proposed in [53] where correlations are extracted from the Web and used by a probabilistic reasoner. Results show that it outperforms that method in recognizing 7 out of 8 ADLs (CASAS dataset).

Inspecting the results of our system, we observe that with the introduction of active learning the recognition rate remains stable or increases. Investigating the results in detail, we notice that the recognition rate of *clean* has a strong increase (ac_7, +17%), while *prepare soup* (ac_6) remains unchanged. A deeper investigation pointed out that activity ac_6 was almost never queried, since its initial semantic correlations derived from our ontology were already sufficient to recognize it accurately. Regarding the other activities, we report an improvement which varies from 1% to 11%.

Considering the individual activities, Figure 5.15 highlights that there are almost no conflicting activity classes and that in general each activity is well recognized. However, we observe that *clean* (ac_7) is often confused with the remaining activities. Indeed, this is because *clean* is not clearly bound to a certain location or sensorized object; hence, during that activity the resident triggers several sensor events that indicate the execution of other activities.

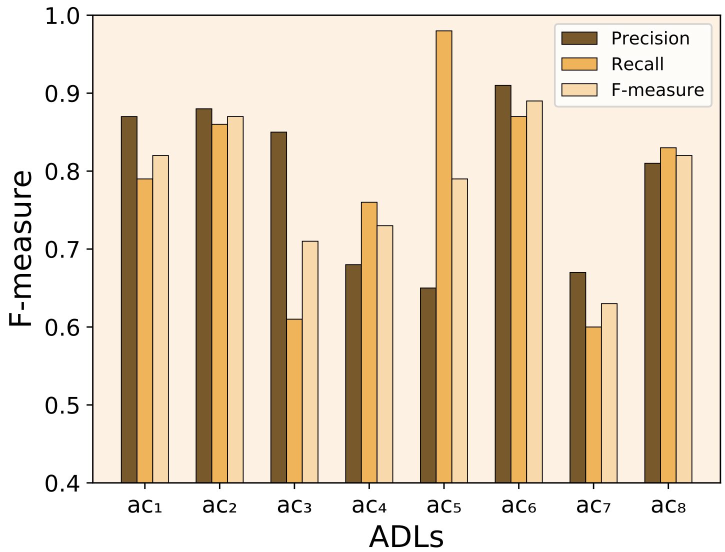


Figure 5.14: Precision, recall and F-measure (with active learning). Entropy threshold $\lambda = 0.9$, feedback support threshold $\sigma = 7.5$

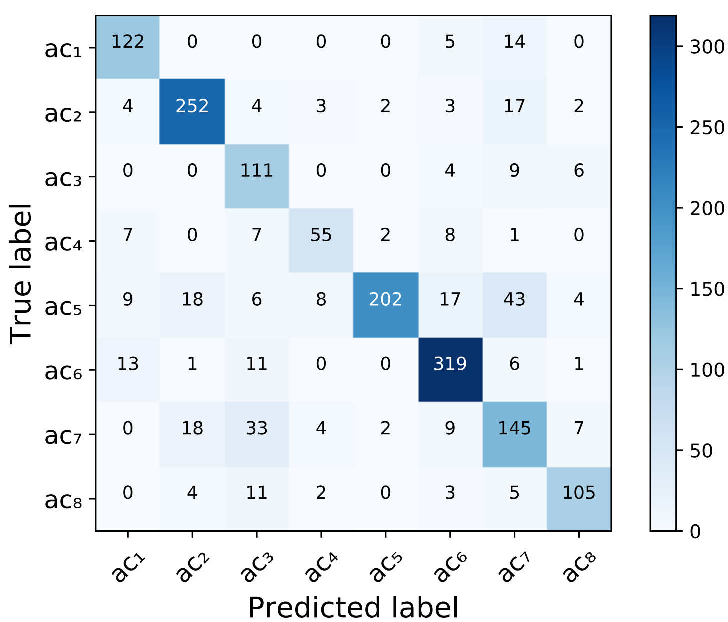


Figure 5.15: Recognizing interleaved ADLs with active learning: Confusion matrix. Entropy threshold $\lambda = 0.9$, feedback support threshold $\sigma = 7.5$

The previous mentioned results were obtained setting the entropy threshold to 0.9. As this value directly influences the number of queries issued by the system, it is an important parameter to consider. Figure 5.16a clarifies that on average a user had to answer six questions to achieve the reported improvement of 5%. In the considered dataset, only one day of ADLs for each subject was available. We expect that the average number of queries in a day for a specific user will significantly decrease over time, thus converging to zero queries after few days. It is important to note that lowering the entropy threshold

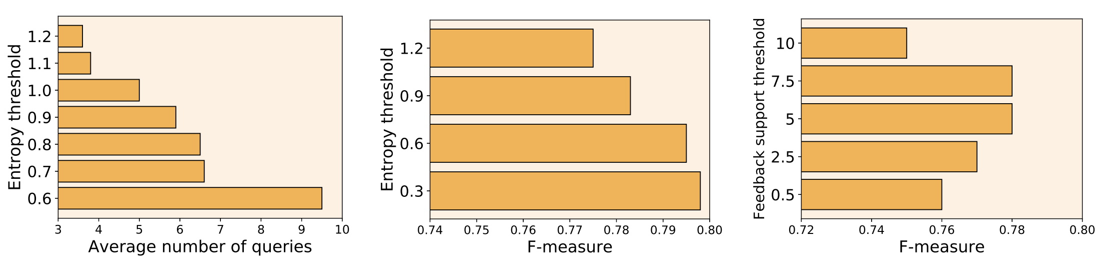
Table 5.11: Results (F-measure) of our system with varying entropy threshold.

Class	**Entropy threshold λ**			
	0.3	**0.6**	**0.9**	**1.2**
ac_1	0.819	0.813	0.824	0.811
ac_2	0.875	0.874	0.869	0.876
ac_3	0.743	0.739	0.709	0.730
ac_4	0.719	0.724	0.724	0.724
ac_5	0.813	0.807	0.784	0.780
ac_6	0.896	0.894	0.887	0.886
ac_7	0.659	0.645	0.633	0.629
ac_8	0.859	0.863	0.824	0.774
avg.	0.798	0.795	0.782	0.776

would still improve our results (see Figure 5.16b) but would determine a significantly higher number of feedback queries (see Figure 5.16a). As expected, we observe a tradeoff between the overall improvement of the recognition rate and the user's effort spent to provide feedback.

Table 5.11 outlines the individual F-measure values that were achieved for each ADL using different values of the entropy threshold. The results confirm that the mentioned tradeoff holds for almost every activity. An exception is *answer the phone* (ac_4) as the recognition rate remains almost unchanged. This can be because the entropy computed on the segments related to this activity is always very high; hence, increasing the entropy threshold does not reduce the number of queries.

In addition to entropy, we also assessed the impact of the feedback support value σ, which ensures that a personalized feedback item is transmitted only if it was derived from a sufficient number of feedback items from similar homes. Figure 5.16c outlines that when σ drops under a certain value, the system uses unreliable feedback, which results in a decreased recognition rate. In contrast, using an excessively large value of σ, the system filters out relevant feedback that could improve recognition rates.



(a) How entropy affects the number of queries

(b) How entropy affects performance (F-measure)

(c) How feedback support affects performance (F-measure)

Figure 5.16: The plots illustrate the relation between our entropy and feedback support thresholds in respect of the recognition quality. Hence, a lower entropy threshold increases the recognition rate but also goes hand in hand with a higher number of questions that the user has to answer (cf. (a) and (b)). In this context, the feedback threshold has to ensure that the unreliable feedback is ignored, i.e., that does not generalize over the group of homes (cf. (c)).

In general, our results clearly show that collaborative active learning is a reliable tool to discover new semantic correlations and in turn to improve the recognition rate. This is especially the case for sensors that do not carry explicit semantic information with respect to activities. For instance, our ontology did not cover the events related to motion sensors. Our system was able to learn automatically the semantic correlation for those sensors' types improving the recognition rate. Moreover, our method required on average only six feedback queries per resident, ranging from a minimum of 3 to a maximum of 10. We believe that this number of questions is acceptable in many application domains, especially if user-friendly and context-aware interfaces for feedback acquisition are used.

5.5 Discussion

Similar to the introduced physical human activity recognition system, there are also technical and conceptual aspects in respect of recognizing ADLs which we need to discuss. First, only few works focus on how to interact with the resident, i.e., to investigate which interface is appropriate to communicate with the user. Probably that depends on the respective person, e.g., one might prefer concrete questions as text or as a voice message while others want to see a picture or a video. Second, so far we only considered the scenario of a single resident; however, a multi-residents scenario is anything but unrealistic. Moreover, it is probably one of the most challenging open issues. Especially the Amazon Go store[6] convey a feeling how difficult it is to track people precisely. Third, regardless of whether video cameras are used or not, the privacy aspect is a very important topic, i.e., for what kind of data is it acceptable to be recorded and how they need to be processed. With regard to the General Data Protection Regulation (GDPR) [193], which was adopted within the European Union, it must be expected that this issue will intensify in the future. Fourth, in our work we mainly rely on ontologies to recognize ADLs. One might argue that constructing an ontology is not worth the effort. Further, those adaptions might be always necessary when applying our presented approach to a new smart-home or environment. In the following, we want to refute these aspects. Finally, as already mentioned in respect of our motivation, the combination of wearable and external sensors might be a promising direction, especially concerning the multi-residents problem. Now that we have investigated both fields, we want to recap this idea.

5.5.1 Interaction with the Residents

In most smart-home scenarios, the interaction or communication with the residents is unavoidable be it to handle uncertainties of the system or to receive and process commands. While more and more publications affirm that it is necessary to keep the user in the loop (e.g. to react to behavior changes) only few works focus on this problem.

[6]Amazon Go is a grocery store where more than 100 cameras were installed to track customers. The idea is to automatically recognize what a person took out of the store. The first store opened in January 2018.

Commonly, researchers assume that the user answers questions of the system always correctly. Moreover, sometimes the user should reconstruct the entire daily routine at the end of the day. In addition, the problem of whether the user has time or mood to answer questions is often ignored. We have to acknowledge that also in our work these aspects were simplified.

Indeed, we only identified few works which deal more closely with this topic. Rashidi et al. [194] present a graphical user interface (named CASA-U) which can be used by the resident to provide explicit and implicit feedback. More precisely, the user can directly manipulate automated activities (e.g. patterns) but also rate them. While the authors state that it was clear to the users how to use CASA-U, they also highlight that training is needed for residents to make effective use of smart-home technologies. In this context, Karami et al. [195] also states that a user-independent scenario is not feasible (also in respect of a multi-residents scenario) but also emphasize that the interaction should be in a natural way without any need to educate or train users. Comparable to CASA-U, they use an interactive tablet combined with a web-based application but consider also voice recognition to gather (explicit and implicit) feedback from the user. Unfortunately, they do not evaluate how well the user interacts with their system but only point out how critical this aspect is. Hossain et al. [161] propose an active learning approach which relies on additional (external) annotators. They compare different algorithms to identify samples that are worse to be labeled. While their results look promising and it is possible to learn new activities at a later point in time, the number of asked queries seem to be unfeasible in respect of a real world scenario. Overall, this shows that there is no straight solution and that it is necessary to dive into this issue. Besides, there are also works which focus in general on the users' behavior in an intelligent environment [196].

In respect of our approach, we believe that it is important to investigate contextual aspects that should be considered when evaluating whether to ask a feedback or not. These aspects include the number of queries that have already been asked recently, the current mood of the subject and whether the user can be interrupted. Regarding the interface, a speech recognition module is probably most promising as it allows to query residents in natural language. Further, a voice interface is particularly suitable for patients and elderly subjects, thus facilitating their interaction with the system. This is also supported by upcoming smart speakers like Google Home, Amazon Echo, and Apples HomePod. Moreover, these devices easily allow to develop and investigate suitable voice interfaces (e.g. as a *Skill* for Alexa which is used by Amazon Echo).

5.5.2 Multiple Residents in a Smart-home

Similar to the preceding problem, recognizing activities in a multi-residents scenario is an open issue and researches still try to identify the most appropriate approach. In this context, researchers mainly try to track and separate the individuals in a smart-home in order to assign sensor events to a certain person or they focus on approaches where a separation of users can be avoided (so both extremes). Further, considering multiple residents also increases the complexity in respect of how activities are performed. Having at least two residents requires (in addition to interleaved activities) also focusing on cooperative, concurrent and parallel activities. Besides, most works state that a camera or video-based solution is not appropriate mainly due to privacy concerns [197–199].

Usually, the choice of approach (i.e. to track or not to track the residents) depends on the considered sensors. On the one hand, some researchers claim that wearable devices are obtrusive [198, 200] while others consider wearable devices as indispensable [52, 201]. Only external sensors seem to be generally accepted. In this context, Alemdar et al. [199] present an approach that considers only ambient sensors and that uses factorial Hidden Markov Models to handle multiple residents at the same time without assuming any explicit user identification. They argue that it is not realistic to assume that a person's identification is available at any time. However, while they considered only two residents, they also conclude that a higher number of residents may require a tracking mechanism as otherwise it would be extremely challenging. Alhamoud et al. [200] have the same goal, i.e., to handle two residents by using only power sensors. They use a multi-label classification approach to reduce the complexity so to avoid a strict assignment of activities to users. Indeed, a multi-label classification approach seem to be a common approach to handle multiple residents [200, 202]. The authors state that the temporal relations between subsequent activities play an essential role in enhancing the predictive performance. Unfortunately, their results have weaknesses, as one activity was not recognized at all. In contrast to these two approaches, Yin et al. [197] also tried to track the residents by just using non-wearable and unobtrusive sensors to localize the residents at room-level using probabilistic models. However, the accuracy is at most 74% (six rooms, two residents).

Researches which propose a combination of external and wearable sensors to track residents, usually also state that the *only-external-sensors* approaches do not scale with regard to a larger number of residents. Roy et al. [52] presents such a hybrid solution using Hidden Markov Models and also investigate the performance concerning a varying number of residents. For that purpose, they rely on spatiotemporal constraints along with multimodal data to recognize postures, locations and events to derive ADLs. Their results show, first that the combination of external and wearable sensors performance most suitable, i.e., better than the respective sensors individually. Second, also the reported performance is stable (accuracy: up to 0.90 ± 0.06) in respect of observing four residents simultaneously. Alam et al. [201] follow a similar idea by also using wearable and external sensors to mine spatio-temporal relationships across the activities of individuals (i.e. constraints and correlations). In this context, they distinguish between micro- (e.g. posture) and macro-activities (e.g. cooking) and use a Bayesian network to derive ADLs. However, the results are less expressive as they consider only two residents. Mokhtari et al. [203] introduce a system that only relies on wearable tags and motion sensors. The wearable tags allow a room-level localization using Bluetooth low energy. Subsequently, the triggered motion sensor events are assigned to a resident. The insights are comparable to the two previous works but the authors highlight and important aspect. The main issue of a wearable-based sensor approach is that people might forget to carry the respective devices or tags.

Overall, it points out that most works only focus on two residents so that the respective results are hard to interpret concerning scalability. It seems that the combination of external and wearable sensors is promising but there is still much research required. This includes the issue how to combine these two sensor types and the derived information. Especially during the last two years, it can be observed that an increasing number of publications focus on multi-residents smart-homes [52, 197–200, 202, 203]; hence, it can be expected that researchers shift from a single-resident to a multi-residents scenario. Indeed, this development is in line with our motivation and our presented work. As intended, we focused on (fundamental) problems that can be considered as prerequisites for the described development. It can be assumed that the individual user will be more and more the *object* of interest (user-centric); thus, wearable and external sensors are required.

5.5.3 Privacy Aspects

Considering smart-homes, usually the term *privacy* goes along with video cameras and computer vision techniques [197, 198]. Researchers argue that cameras record events in a very detailed way, which on the one hand include unnecessary privacy details while on the other hand there is also the danger that the cameras are controlled (take-over) by a third-person. However, the term *privacy* is actually much broader and includes also concerns about data transfer and the question if even the recorded sensor data is private data. Indeed, the necessity of discussing these questions goes hand in hand with the considered approach, i.e., should the ADL recognition within a smart-home run independently, collaborate with other smart-homes, or make us of external service providers. The last two raise many privacy concerns even if the proposed system does not require cameras or microphones.

For the sake of our work, we assumed that the introduced Cloud Service (see Section 5.3.3.1) is trusted, while in a real scenario it can be considered as an untrusted third party service. Hence, there is also the need of protecting the confidentiality and integrity of user and infrastructure profiles but also information about events and activities provided by the user feedback. We believe that a solution based on homomorphic encryption [204] and secure multi-party computation [205] may sufficient to address the outlined problems. Thus, recorded data is encrypted before it is transmitted to the Cloud Service that in turn is able to perform the required computations without encrypting the data. While such techniques exist, the feasibility in respect of collaborative smart-home scenarios is still unclear, i.e. this can be considered as an open issue.

Another important aspect which goes along with this topic is the GDPR [193]. The GDPR is a regulation adopted by the European Union (EU) in mid-2018 for privacy and data protection for all individuals within the EU. Broadly speaking, the GDPR aims to give control to the individuals over their personal data but also to control export of personal data outside the EU. This means that in addition to the already mentioned issues, also transparency is an important factor. The individuals need to know which data is recorded and how this data is used. Further, it can be assumed that in the medium term also other countries adopt similar rules.

While we acknowledge that privacy is a critical aspect, we believe it should not dominate the investigations of open issues. On the one hand, it should be considered in the design of new system architectures but also regarding the feasibility in a real-world scenario. On the other hand, new questionable ideas or approaches should not be discarded immediately.

5.5.4 Ontology Engineering

Ontologies enable to define concepts and relationships between those concepts within a domain. In this context, ontology engineering means the modeling of a large-scale representation of corresponding actions, time aspects, physical objects and beliefs. Especially in the last decade, the use of ontologies in information systems has become more and more popular in various fields including web technologies and natural language processing [206].

In our work, we made use of ontologies to define formally the semantics of ADLs, sensor events, context data, and the home environment. The reason was to overcome the issue that manually modeling these things is unfeasible in realistic scenarios. For instance, the CASAS dataset (see Section 5.1.1) which we used in our experiments involves 70 sensors and 8 activities, resulting in 560 different values of *semantic correlations*. Other real-world deployments are much more complex. Of course, we acknowledge that our technique requires a relevant knowledge engineering effort to define the required ontology (our ontology includes 235 classes and 59 properties). However, we point out that the knowledge engineering effort can be reduced by reusing existing ontologies. In particular, the ontology used in this work is an extension of the COSAR ontology [145], which was originally intended to model context data and human activities. The extension mainly regarded the definition of a few classes for activities and artifacts that were not considered before, and a few additional properties used by our reasoning method. Developing the extension required one day of work by a researcher with good skills in OWL 2 modeling. Moreover, we were able to use the same ontology for both apartments involved in our first experiments, which had very different characteristics (see Sections 5.1.1 and 5.1.2).

We agree that it is questionable whether in larger scale implementations the same ontology can be adequate to cover every possible home environment and individuals' mode of activity execution. That is why we also exploited active learning to fine-tune the probabilistic model according to the user's environment and personal habits, and to evolve automatically the ontology according to the current context. Nevertheless, we have to conclude that even if our system relies on a generic and possibly incomplete ontology that considers (only) general relationships between activities and home infrastructure, the engineering effort is still noticeable. On the other hand, even if this is not an optional solution it surpasses manually modeling and offers many benefits.

5.5.5 User-Centric Activity Recognition

As previously mentioned, by *user-centric* we denote an approach which combines wearable and external sensors for recognizing ADLs, i.e., a hybrid solution. In this context, we make no assumptions about the sensors used, how they are combined, or the location of the user. However, we believe that it is necessary to identify users in respect of being able to assign the triggered or recorded sensor events to the respective user. Indeed, we already discussed this approach in respect of a multi-residents scenario and the related works present promising results (see Section 5.5.2). However, we take the view that the combination of wearable and external sensors is already meaningful concerning a single resident as for example the observation of the arm movements might clarify if a phone is just touched or actually used (cf. see Section 5.4.1.1). Further, we also believe that the recognition of ADLs should not be restricted to a certain environment. Of course, leaving a smart-home goes along with losing information which is provided by external sensors. However, upcoming devices such as smart-glasses might be a bridge solution. Certainly, the biggest issue is probably the social acceptance of such a device. In contrast, simplified cameras which only capture depth or brightness information might be a tradeoff. Overall, we consider our presented work on the one hand as essential in regard of the outlined research directions on the other hand existing related work that focuses on hybrid solutions and multimodal data confirms our statements.

Compared to the multi-residents discussion, below we want to focus more on the combination or fusion of different sensors and the resulting issues. Basically, one distinguishes between early and late fusion, i.e., the recorded data is fused before the actual machine learning technique is applied or the different sensor streams are processed (e.g. classified) separately and the individual results are subsequently combined. Existing works tend to use late fusion not least because problems with varying sensor sampling can be avoided (e.g. video vs. acceleration data [207]). Alam et al. [201] present such a hybrid solution which makes use of late fusion. They focused on recognizing context data including posture, location and environmental noise to recognize complex activities. Similar, Wang et al. [208] rely on distributed ambient sensors to identify the current room of the user. Subsequently, they analyze the wearable sensors to derive the performed ADL. This two-step approach also enables to incorporate certain constraints, as for example it is impossible to cook in a bathroom. They state that single sensor modalities sometime may not cope with complex situations in practice. Further, De et al. [107] also states that ADLs often include physical and postural activities while IADLs (see Section 1.1) include activities that require a combination of physical and cognitive capabilities. While it is possible to capture such aspects only using external sensors, wearable sensors are also capable of this but in a much simpler way. For instance, a single accelerometer which is attached to a forearm can capture how someone is moving the arm. In contrast, especially without cameras one needs a variety of external sensors to capture the movement of a certain arm. This also shows that combining external and wearable sensors goes along with a reduced infrastructure, which is also a common goal or requirement.

Generalizing the existing late fusion approaches, it strikes that the idea is to recognize certain (critical) parts like object interactions, physical activities, current location, and (emotional) conditions which in turn are combined to recognize the performed ADL [107, 197,201,208] (see Figure 6.1). However, even though this general approach seems suitable for recognizing ADLs, there are several open issues. This includes the question how different aspect or context information (e.g. posture) contribute to the final decision but also how fine granular this aspects need to be recognized. Hence, this approach has many steep operational challenges. In this context, Roy et al. [52] also highlights that usually individuals appear reluctant to wear continually multiple sensors on the body. Further, embedding sensors on various objects of daily living (e.g. microwaves and kitchen cabinets) also go along with operational costs and battery-life issues.

Nevertheless, we believe that these problems can be solved. First, upcoming solutions like smart-clothes [63] may change the acceptance of carrying sensors permanently. Besides, even if several researchers state that carrying wearable devices is disturbing especially for elderly people, at least in Germany it is common that elderly people have an emergency call system. Thus, these people wear an emergency button all the time (e.g. as a necklace or bracelet). Moreover, in our presented work, we already addressed and discussed several issues and the results show evidence for the feasibility in a real-world scenario.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Human Activity Recognition has been deeply investigated in the last decade taking advantage of the effective sensing infrastructure that is becoming available with off-the-shelf products as part of domotics, smart objects and wearable devices. However, a general problem of many existing studies on the subject is that they are conducted in a highly controlled environment. In consequence, the results of these studies often do not carry over to real world applications. In our work, we investigated sensor-based human activity recognition with the objective of moving out of the laboratory but also of creating a basis for combining wearable devices and smart-environments. Overall, we addressed several open issues and proved the feasibility of our introduced solutions but at the same time, we also identified further research directions. In the following, we will go into detail and recap our research questions and the respective results.

Our first investigation focused on an outstanding problem when relying on wearable devices. This is the fact that it is up to the user where the device is carried, i.e. the on-body device position is not known a-prior (**RQ1.1**). In contrast, most existing works assume to know the device position. To dive into that problem, we created a large real world dataset by recording 7 on-body positions of 15 subjects while they performed eight physical activities. Considering a single-subject scenario, we investigated the possibility to detect the current on-body position of a wearable device in a real world scenario with a single accelerometer in context of several physical activities. Our results show that we are able to detect the correct on-body device position with 89% (F-measure). Further, we want to highlight that the recognition quality of the device position was almost stable (F-measure, SD ±3.4%). Considering the individual physical activities, standing and sitting are the most problematic where jumping and running are the most appropriate ones.

In addition, to evaluate the impact of the position information, we performed position-aware activity recognition experiments where we considered the results of the on-body position detection including all mistakes (**RQ1.2**). The corresponding results show that the introduced position-aware approach is able to recognize the correct physical activity with 84% (F-measure). Compared to the position-independent approach, the recognition rate is 4% higher, i.e., the results provide strong evidence for the improvement of the activity recognition rate in case that the on-body position is known.

Other researchers achieved lower or equivalent recognition rates and considered less positions and activities. For instance, Coskun et al. [18] considered the hand, trousers, and backpack and achieved a recognition rate of 85%. Furthermore, Vahdatpour et al. [118] considered the same on-body positions as we did expect the chest and focused only on *walking* but achieved an accuracy of 89%. This indicates that the consideration of more positions and activities lead to a lower recognition rate. This is in line with our first result were we did not distinguish between static and dynamic activities. However, due to the individual handling of dynamic and static activities, our introduced approach performs significantly better in a real world scenario.

Equally important, Coskun et al. [18] state that the usefulness of the information of the device position depends on the performed activity. Further, they also state that in general this information has a less effect on the recognition rate. In contrast, Martin et al. [117] state that the information of the position leads to a significant improvement concerning the activity recognition. In view of the fact that we considered all relevant on-body positions and several different and common physical activities, our results also provide strong evidence concerning the positive influence of the position information.

As a single-subject scenario goes hand in hand with the need of labeled data for each user but people like patients or elderly may unable to do that, consequently we focused on the feasibility of using labeled data across people (**RQ1.3**). In particular, we investigated the following approaches: leave-one-subject-out, random groups, top-pairs, and grouping people with similar physical characteristics. The results show that our physical-based recognition model performs the best, i.e., physical characteristic (fitness level, body structure, and gender) enable to build promising cross-subjects activity recognition models. Further, our results also show that the waist is the best on-body position for cross-subjects activity recognition. Hence, acceleration patterns for the same activity across different users are most similar at this position. Considering this position, the physical-based approach was able to achieve a recognition rate of 79%. With an additional wearable device (at the shin), the recognition rate improves by +3% (82%).

Most existing works focus on leave-one-subject-out where the opinions tend to state that this approach is not reliable. Vo et al. [135] clarify that an increasing number of considered users goes along with a decreasing activity recognition performance. We attribute this behavior to the fact that the classifier learns only the most dominant behaviors across people. To counteract this behavior, researchers suggest creating specific groups. In particular, Lara et al. [108] and Weiss et al. [122] hypothesize that physical characteristics such as gender, weight, and fitness level could be reliable indicators to form groups. In our work, we investigated this hypothesis and our results provide evidence for the correctness. However, we also have to state that the considered physical characteristics did not cover the characteristics of the activity jumping.

Comparing our single-subject and cross-subjects results, shows a performance gap but also the shortcoming that our approach is not capable to adapt to behavioral changes of the user (**RQ1.4**). For that purpose, we investigated the possibility of personalizing cross-subjects activity recognition models using an Online Random Forest (as in preceding experiments it turned out to perform consistently better than other classification techniques). Similar to the preceding experiments, we considered all on-body device positions but also combinations and focused on physical activities. The results show that by relying on user-feedback and smoothing, the recognition rate for a new unseen subject can be improved by +8% while dynamic activities (which are normally of higher interest) can be even improved by +11% (F-measure). Hence, online and active learning are suitable techniques for increasing significantly the recognition rate of a cross-subjects based model. The resulting effort for the target user that goes along with the personalization was limited to 10 questions, i.e., significantly less effort than creating and labeling a new dataset.

In regard of related work, we can state that our approach achieves a higher improvement than a combination of neural networks and fuzzy clustering [132] or online parameter optimization [39, 130]. Further, related work also suggests that an extension of our approach by co-training could be a promising idea [126].

To answer **RQ2.1** and **RQ2.2**, we performed an extensive literature research which in turn lead us to an unsupervised approach for recognizing complex ADLs through ontological and probabilistic reasoning with Markov Logic Networks. Extensive experiments with real-world datasets showed that the accuracy of our unsupervised method is comparable to the one of supervised approaches, even using a smaller number of sensors. For instance, compared to Singla et al. [186], our approach performed +8% better (F-measure, CASAS). On the negative side, our technique requires a relevant knowledge engineering effort to define a comprehensive ontology of ADLs, the home environment, and sensor events. However, the ontology used in this work is an extension of the COSAR ontology [145], which was originally intended to model context data and human activities. Hence, it is feasible to use the same ontology across different works with a manageable effort. Nevertheless, the modeling problem is particularly challenging when focusing on complex ADLs, which are characterized by large intra- and inter-personal variability of execution as it is unfeasible to model manually these aspects in realistic scenarios. For instance, the CASAS dataset that we used in our experiments involves 70 sensors and 8 activities, resulting in 560 different values of *semantic correlations*. Other real-world deployments are much more complex. For that reason, we state that our approach is a suitable tradeoff between engineering effort, feasibility and scalability because it overcomes several limitations including the need to acquire expensive ADL datasets, enumerating all possible sequences of actions, and it can be seamlessly reused.

Another major concern is the ability to recognize ADLs also shortly after or even during the execution; hence, similar to the physical activity recognition scenario, several applications require recognizing ADLs in almost real-time (**RQ2.3**). That also implies the question how to process or segment the sensor stream. For that reason, we enhanced our introduced system by a novel online segmentation algorithm that combines probabilistic and symbolic reasoning to segment on the fly the continuous stream of sensor events. More precisely, we considered different aspects such as *object interaction*, *change of context*, *consistency likelihood*, *time leap*, and *change of location* that can be directly derived from the sensor stream. For both datasets, the experiments show that our segmentation algorithm produces high quality segments with respect to standard techniques, enabling to reach higher recognition rates. In this context, the individual segments are processed and classified in the same way as in offline mode, i.e. through ontological and probabilistic reasoning. Comparing both modes, the recognition rate achieved in online mode (76%, CASAS) is close to the one achieved in offline mode (78%, CASAS).

The capability of being able to recognize ADLs in almost real-time benefits to gather the behavioral pattern of the user (**RQ2.4**). Thus, we focused on a concept which takes advantage of the heterogeneity of environments and individuals to discover new *semantic correlations* between certain sensor events and ADLs. Experimental results show that our framework significantly improves the overall recognition rate (+5%, F-measure), while issuing a limited number of queries to the inhabitants. Further, we observe a tradeoff between the overall improvement of the recognition rate and the user's effort spent to provide feedback. In order to compare our approach with state-of-the-art techniques, we also implemented a supervised method [192], which relies on machine learning and time-based feature extraction. Results show that our approach outperforms the supervised method in terms of average F-measure ($74 \pm 15\%$ vs. $78 \pm 9\%$, F-measure, CASAS). We also compared our approach with a recent unsupervised method proposed by Riboni et al. [53] where correlations are extracted from the Web and used by a probabilistic reasoner. Results show that it also outperforms that method ($70 \pm 20\%$ vs. $78 \pm 9\%$, F-measure, CASAS). In general, our results clearly show that collaborative active learning is a reliable tool to discover new *semantic correlations* and in turn to improve the recognition rate. This is especially the case for sensors that do not carry explicit semantic information with respect to activities. For instance, our ontology did not cover the events related to motion sensors but our system was able to learn automatically the *semantic correlation* for those sensors' types improving the recognition rate.

Overall, the answers of our research questions illustrate on the one hand the feasibility of physical activity recognition but also recognizing ADLs in a real world scenario. On the other hand, we also identified other open issues or even limitations. This brings us to one of our core motivations, i.e. to pave the way for combining external and wearable sensors. In respect of our comprehensive experiments and discussions, we conclude that this is a promising way to overcome several discussed issues. Indeed, already existing but limited hybrid solutions show that it is reasonable to dive into this concept. Figure 6.1 summaries our work, our ideas, and approach in respect of considered aspects, how they are connected, what we already investigated, and what else needs to be investigated.

6.2 Future Work

There are several open issues for physical activity recognition, recognizing ADLs in a smart-environment but also in respect of hybrid solutions. For that reason, in the following we highlight certain research directions in respect of these areas where we mainly refer to our preceding discussions.

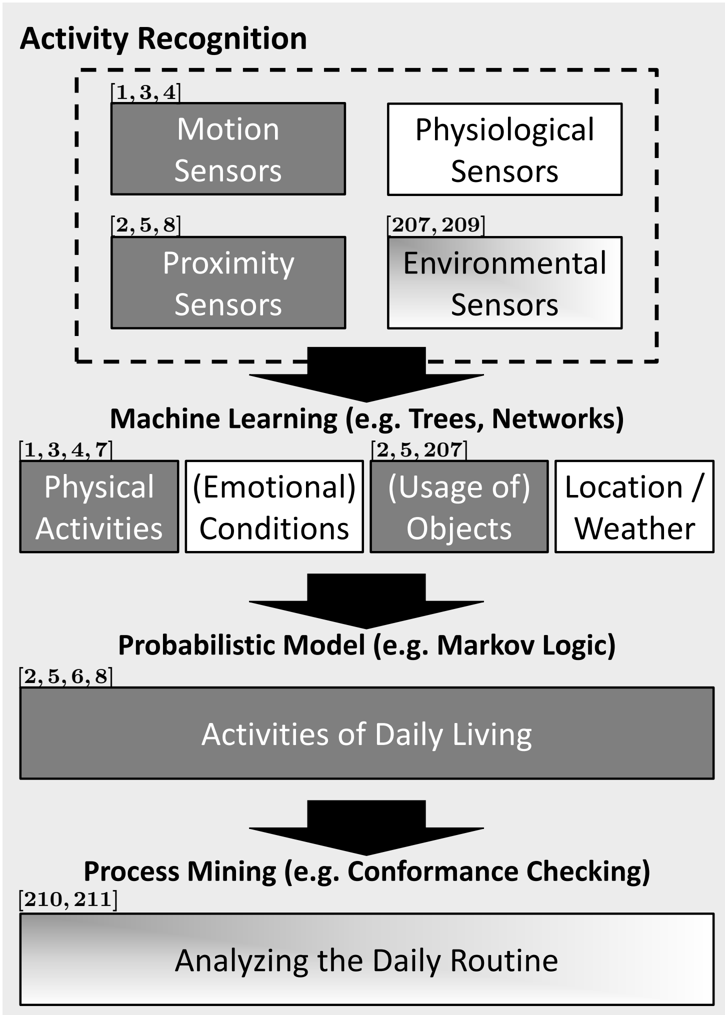


Figure 6.1: Towards real world activity recognition from external and wearable sensors (adapted from [212]). It depicts the overall picture which we have in mind when talking about activity recognition. More precisely, the picture explains the flow from raw sensor signals to certain aspects like the current posture (e.g. standing) or used objects (e.g. knife) which in turn enable to derive the performed ADL (e.g. preparing meal). After recognizing a sequence of ADLs, they can be connected to analyze the (daily) routine. The grey boxes highlight areas which we already investigated; hence, the depict references are our publications. The grey/white boxes indicate that we performed only basic investigations in that corresponding field.

So far, we have shown that physical activity recognition based on wearable devices can be reliably executed in a real world setting and the necessary training effort can be reduced significantly using online and active learning. Nevertheless, we only focused on accelerometers where wearable devices provide several different kind of sensors. While several works already considered, for example, gyroscopes and magnetometers, they did not investigate how certain sensors contribute to the recognition results. Further, wearable devices like smart-watches seem to be predestined to recognize sedentary activities; indeed, this can be also considered as a step from physical activities to ADLs. For instance, first it is recognized that someone is sitting and subsequently the arm movements are observed to recognize the actual ADL (e.g. eating). On the other side, we also have the methodological part. In our work, we relied mainly on Random Forest. Upcoming or hyped techniques like LightGBM or deep neural networks might even improve the performance but in particular, it is not clear which open issue could be addressed with these techniques.

In respect of recognizing ADLs, we have proposed purely unsupervised methods for recognizing high-level activities; however, these approaches were tested in a partly restricted setting. This includes the number of residents, the interaction with the residents, but also privacy aspects. A multi-inhabitant scenario introduce several issues such as the belonging of the sensor events, i.e., which user triggered which sensor event, but also new ways of carrying out activities must be taken into account, e.g., parallel or cooperative activities. In particular, wearable devices seem to be promising in respect of these issues. Further, the interaction with the inhabitants is not limited to the type of communication interface (e.g. voice). An important aspect that also needs to be considered is the current context and mood of the user, which may influence the quality of the answer and the willingness to provide an answer. The definition of user-friendly interfaces for that purpose is also a challenging aspect, which needs to be investigated.

Finally, especially in respect of the GDPR, the privacy issue has a significant influence on which sensors or devices should or can be considered but also how the recorded data need to be processed. This includes necessity of transparency that in turn also may have an influence on the considered method, i.e. the result has to be comprehensible and the corresponding model explainable.

Appendices

Appendix A

Authors' Contribution

The following tables clarify the contribution of each author in respect of the publications that were considered in this work. The order of the contribution and keywords is arbitrary.

Table A.1: On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition (2016) [1]

Authors	Contribution	Keywords
Timo Sztyler	Methodology Writing (original draft) Data curation Investigation/Experiments	Data collection Position detection approach Position-aware HAR
Heiner Stuckenschmidt	Writing (review/editing) Conceptualization Resources Supervision	Introduction Related work Experimental design

Table A.2: Unsupervised Recognition of Interleaved Activities of Daily Living through Ontological and Probabilistic Reasoning (2016) [2]

Authors	Contribution	Keywords
Daniele Riboni	Methodology Writing (original draft) Conceptualization Supervision	Description logic/Ontology Architecture Semantic correlation reasoner Introduction
Timo Sztyler	Methodology Data curation Investigation/Experiments Writing (original draft)	MLN modeling Data analysis MAP inference
Gabriele Civitarese	Methodology Data curation Investigation/Experiments Writing (original draft)	Statistical analysis of events MLN knowledge base Implementation
Heiner Stuckenschmidt	Writing (review/editing) Conceptualization Resources Supervision	Preliminaries Discussion

Table A.3: Position-Aware Activity Recognition with Wearable Devices (2017) [3]

Authors	Contribution	Keywords
Timo Sztyler	Methodology Writing (original draft) Data curation Investigation/Experiments	Cross-subjects approaches Physical characteristics Multi-sensor setup
Heiner Stuckenschmidt	Writing (review/editing) Conceptualization Resources Supervision	Introduction Experimental design
Wolfgang Petrich	Project administration	-

	Funding acquisition	

Table A.4: Online Personalization of Cross-Subjects based Activity Recognition Models on Wearable Devices (2017) [4]

Authors	Contribution	Keywords
Timo Sztyler	Methodology Writing (original draft) Data curation Investigation/Experiments	Online Random Forest User-feedback/smoothing
Heiner Stuckenschmidt	Writing (review/editing) Conceptualization Resources Supervision	Introduction Experimental design

Table A.5: NECTAR: Knowledge-based Collaborative Active Learning for Activity Recognition (2018) [5]

Authors	Contribution	Keywords
Gabriele Civitarese	Methodology Conceptualization Investigation/Experiments Data curation Writing (original draft)	Architecture Query decision Implementation
Claudio Bettini	Methodology Conceptualization Formal analysis Writing (original draft) Validation	Collaborative feedback agg. Semantic correlation updater Introduction
Timo Sztyler	Data curation Investigation/Experiments Writing (original draft) Validation	Online rule-based segm. MLN modeling
Daniele Riboni	Methodology Conceptualization Writing (original draft)	Ontological model Related work
Heiner Stuckenschmidt	Writing (review/editing) Resources Supervison	Discussion Experimental design

Table A.6: Modeling and reasoning with Problog: An application in recognizing complex activities (2018) [6]

Authors	Contribution	Keywords
Timo Sztyler	Methodology Writing (original draft) Data curation Investigation/Experiments	Problog usage Problog modeling Introduction
Gabriele Civitarese	Data curation Writing (original draft) Conceptualization Investigation/Experiments	Implementation Experimental design
Heiner Stuckenschmidt	Writing (review/editing) Conceptualization	Discussion

	Resources Supervision	

Table A.7: Hips Do Lie! A Position-Aware Mobile Fall Detection System (2018) [7]

Authors	**Contribution**	**Keywords**
Christian Krupitzer	Methodology Writing (original draft) Investigation/Experiments Conceptualization	Self-Adaptive Fall Detection Related Work MAPE cycle
Timo Sztyler	Writing (original draft) Data curation Investigation/Experiments Conceptualization	Cross-Datasets Fall Detection Position-Aware Fall Detection Introduction
Janick Edinger	Writing (original draft) Data curation Investigation/Experiments	Data preparation Experimental design Discussion
Martin Breitbach	Writing (original draft) Data curation	Literature research Implementation
Heiner Stuckenschmidt	Writing (review/editing) Resources Supervision	Reviewing
Christian Becker	Writing (review/editing) Resources Supervision	Reviewing

Table A.8: POLARIS: Probabilistic and Ontological Activity Recognition in Smart-homes (2019) [8]

Authors	**Contribution**	**Keywords**
Gabriele Civitarese	Methodology Writing (original draft) Investigation/Experiments Data curation	Architecture Statistical analysis of segments Segmentation evaluation
Timo Sztyler	Methodology Writing (original draft) Investigation/Experiments	Online segmentation MLN modeling Experimental design
Daniele Riboni	Methodology Writing (original draft) Conceptualization	Description logic/Ontology Semantic correlation reasoner Related work
Claudio Bettini	Writing (review/editing) Conceptualization Supervision	Discussion Introduction
Heiner Stuckenschmidt	Writing (review/editing) Resources Supervision	Reviewing

Appendix B

Sensor Feature Framework

The following formulas illustrate how the respective features were implemented in our provided *Sensor Feature Extraction* framework.

Mean

$$\bar{x} = \frac{1}{n} * \sum_{i=1}^{n} x_i \tag{B.1}$$

Variance

$$var(x) = \frac{1}{n} * \sum_{i=1}^{n} (\bar{x} - n_i)^2 \tag{B.2}$$

Standard Deviation

$$\sigma_x = \sqrt{var(x)} \tag{B.3}$$

Interquartile Range (type R-5)

$$\begin{gathered} iqr = Q_{0.75} - Q_{0.25} \\ Q_p = x_{\lfloor h \rfloor} + (h - \lfloor h \rfloor) * (x_{\lfloor h \rfloor + 1} - x_{\lfloor h \rfloor}) \\ h = Np + \frac{1}{2} \\ where\ \forall x_i, x_j \in X x_i \leq x_j\ and\ i \leq j \end{gathered} \tag{B.4}$$

Mean absolute deviation

$$mad = \frac{1}{n} * \sum_{i=1}^{n} |x_i - \bar{x}| \tag{B.5}$$

Kurtosis

$$w = \frac{1}{n} * \sum_{i=1}^{n} (\frac{x_i - \bar{x}}{\sigma_x})^4 \tag{B.6}$$

Energy (Fourier, Parseval)

$$\begin{gathered} Energy(Y) = \frac{1}{n} * \sum_{i=1}^{n} (F_i)^2 \\ where\ F_i\ is\ the\ i\text{-}th\ component\ of\ the\ Fourier\ Transform\ of\ Y \end{gathered} \tag{B.7}$$

Correlation Coefficient (Pearson)

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} * \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{B.8}$$

Entropy (Shannon)

$$\begin{aligned} IG(S,F) &= E(S) - \sum_{v \in Values(F)} \frac{|S_v|}{S} * E(S_v) \\ where\ S_v &= \{s \in S | F(s) = v\} \\ E(S) &= -\sum_{i=1}^{|C|} P(i) * log_2(P(i)) \end{aligned} \tag{B.9}$$

$where\ P(i)\ is\ the\ fraction\ of\ examples\ in\ S\ which\ is\ assigned\ by\ label\ c_i$

Median

$$\bar{a} = \begin{cases} x_{\frac{n+1}{2}} & \text{n odd} \\ \frac{1}{2} * (x_{\frac{n}{2}} + x_{\frac{n}{2}+1}) & \text{n even} \end{cases} \tag{B.10}$$

$where\ \forall x_i, x_j \in X x_i \leq x_j\ and\ i \leq j$

Bibliography

[1] T. Sztyler and H. Stuckenschmidt, "On-body localization of wearable devices: An investigation of position-aware activity recognition," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE Computer Society, 2016, pp. 1–9, doi: 10.1109/PERCOM.2016.7456521.

[2] D. Riboni, T. Sztyler, G. Civitarese, and H. Stuckenschmidt, "Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing.* ACM, 2016, pp. 1–12, doi: 10.1145/2971648.2971691.

[3] T. Sztyler, H. Stuckenschmidt, and W. Petrich, "Position-aware activity recognition with wearable devices," *Pervasive and Mobile Computing*, vol. 38, no. Part 2, pp. 281–295, 2017, doi: 10.1016/j.pmcj.2017.01.008.

[4] T. Sztyler and H. Stuckenschmidt, "Online personalization of cross-subjects based activity recognition models on wearable devices," in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE Computer Society, 2017, pp. 180–189, doi: 10.1109/PERCOM.2017.7917864.

[5] G. Civitarese, C. Bettini, T. Sztyler, D. Riboni, and H. Stuckenschmidt, "NECTAR: Knowledge-based collaborative active learning for activity recognition," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE Computer Society, 2018, pp. 125–134, doi: 10.1109/PERCOM.2018.8444590.

[6] T. Sztyler, G. Civitarese, and H. Stuckenschmidt, "Modeling and reasoning with ProbLog: An application in recognizing complex activities," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops).* IEEE Computer Society, 2018, pp. 781–786, doi: 10.1109/PERCOMW.2018.8480299.

[7] C. Krupitzer, T. Sztyler, J. Edinger, M. Breitbach, H. Stuckenschmidt, and C. Becker, "Hips do lie! A position-aware mobile fall detection system," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE Computer Society, 2018, pp. 95–104, doi: 10.1109/PERCOM.2018.8444583.

[8] G. Civitarese, T. Sztyler, D. Riboni, C. Bettini, and H. Stuckenschmidt, "POLARIS: Probabilistic and ontological activity recognition in smart-homes," *Under Review*, vol. tba, no. tba, p. tba, 2019, doi: tba.

[9] P. Dawadi, D. J. Cook, and M. Schmitter-Edgecombe, "Automated cognitive health assessment using smart home monitoring of complex tasks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 6, pp. 1302–1313, 2013, doi: 10.1109/TSMC.2013.2252338.

[10] C. Parsey and M. Schmitter-Edgecombe, "Applications of technology in neuropsychological assessment," *The Clinical Neuropsychologist*, vol. 27, no. 8, pp. 1328–1361, 2013, doi: 10.1080/13854046.2013.834971.

[11] C. Griffiths, C. Rooney, and A. Brock, "Leading causes of death in England and Wales – How should we group causes?" *Health statistics quarterly / Office for National Statistics*, vol. 28, no. 9, pp. 6–17, 2005.

[12] D. Selbie and J. Newton, "Health profile for England: Telling a story about our health," *Public Health England*, 2017.

[13] N. Davies, D. P. Siewiorek, and R. Sukthankar, "Activity-based computing," *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 20–21, 2008, doi: 10.1109/MPRV.2008.26.

[14] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui, "SmartFABER: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment," *Artificial Intelligence in Medicine*, vol. 67, no. Supplement C, pp. 57–74, 2016, doi: 10.1016/j.artmed.2015.12.001.

[15] F. Foerster, M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: A validation study in ambulatory monitoring," *Computers in Human Behavior*, vol. 15, no. 5, pp. 571–583, 1999, doi: 10.1016/S0747-5632(99)00037-0.

[16] M. Chan, D. Estève, C. Escriba, and E. Campo, "A review of smart homes - present state and future challenges," *Computer methods and programs in biomedicine*, vol. 91, no. 1, pp. 55–81, 2008, doi: 10.1016/j.cmpb.2008.02.001.

[17] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "CASAS: A smart home in a box," *Computer*, vol. 46, no. 7, pp. 62–69, 2013, doi: 10.1109/MC.2012.328.

[18] D. Coskun, O. D. Incel, and A. Ozgovde, "Phone position/placement detection using accelerometer: Impact on activity recognition," in *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. IEEE Computer Society, 2015, pp. 1–6, doi: 10.1109/ISSNIP.2015.7106915.

[19] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber, "Smartwatch-based activity recognition: A machine learning approach," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics*. IEEE Computer Society, 2016, pp. 426–429, doi: 10.1109/BHI.2016.7455925.

[20] M. Stikic, T. Huynh, K. Van Laerhoven, and B. Schiele, "ADL recognition based on the combination of RFID and accelerometer sensing," in *Second International Conference on Pervasive Computing Technologies for Healthcare*. IEEE Computer Society, 2008, pp. 258–263, doi: 10.1109/PCTHEALTH.2008.4571084.

[21] M. P. Lawton and E. M. Brody, "Assessment of older people: Self-maintaining and instrumental activities of daily living," *The Gerontologist*, vol. 9, no. 3 Part 1, pp. 179–186, 1969, doi: 10.1093/geront/9.3_Part_1.179.

[22] D. Foti and J. S. Koketsu, *Activities of Daily Living*, 7th ed. Elsevier, 2013, ch. Part III, pp. 157–232. ISBN 978-0323059121

[23] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, "Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor," in *2007 11th IEEE International Symposium on Wearable Computers*. IEEE Computer Society, 2007, pp. 37–40, doi: 10.1109/ISWC.2007.4373774.

[24] N. Roy, A. Misra, and D. Cook, "Infrastructure-assisted smartphone-based ADL recognition in multi-inhabitant smart environments," in *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE Computer Society, 2013, pp. 38–46, doi: 10.1109/PerCom.2013.6526712.

[25] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*. AAAI Press, 2005, pp. 1541–1546.

[26] Y. Jia, "Diatetic and exercise therapy against diabetes mellitus," in *2009 Second International Conference on Intelligent Networks and Intelligent Systems*. IEEE Computer Society, 2009, pp. 693–696, doi: 10.1109/ICINIS.2009.177.

[27] A. D. Association *et al.*, "Diagnosis and classification of diabetes mellitus," *Diabetes Care*, vol. 33, no. Supplement 1, pp. 62–69, 2010, doi: 10.2337/dc10-S062.

[28] W. H. Organization, *Global Report On Diabetes*, 1st ed. World Health Organization, 2016. ISBN 978-9241565257

[29] P. E. Cryer, S. N. Davis, and H. Shamoon, "Hypoglycemia in diabetes," *Diabetes Care*, vol. 26, no. 6, pp. 1902–1912, 2003, doi: 10.2337/diacare.26.6.1902.

[30] J. K. Snell-Bergeon and R. P. Wadwa, "Hypoglycemia, diabetes, and cardiovascular disease," *Diabetes Technology & Therapeutics*, vol. 14, no. S1, pp. S–51–S–58, 2012, doi: 10.1089/dia.2012.0031.

[31] S. R. Colberg, R. J. Sigal, J. E. Yardley, M. C. Riddell, D. W. Dunstan, P. C. Dempsey, E. S. Horton, K. Castorino, and D. F. Tate, "Physical activity/exercise

and diabetes: A position statement of the American diabetes association," *Diabetes Care*, vol. 39, no. 11, pp. 2065–2079, 2016, doi: 10.2337/dc16-1728.

[32] T.-H.-C. Nguyen, J.-C. Nebel, F. Florez-Revuelta *et al.*, "Recognition of activities of daily living with egocentric vision: A review," *Sensors*, vol. 16, no. 1, p. 72, 2016, doi: 10.3390/s16010072.

[33] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing: Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria, April 21-23, 2004. Proceedings.* Springer, 2004, pp. 1–17, doi: 10.1007/978-3-540-24646-6_1.

[34] L. Atallah, B. Lo, R. King, and G.-Z. Yang, "Sensor positioning for activity recognition using wearable accelerometers," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, no. 4, pp. 320–329, 2011, doi: 10.1109/TBCAS.2011.2160540.

[35] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011, doi: 10.1145/1964897.1964918.

[36] L. Barkhuus and V. E. Polichar, "Empowerment through seamfulness: Smart phones in everyday life," *Personal and Ubiquitous Computing*, vol. 15, no. 6, pp. 629–639, 2011, doi: 10.1007/s00779-010-0342-4.

[37] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014, doi: 10.1145/2499621.

[38] D. Guan, W. Yuan, Y. K. Lee, A. Gavrilov, and S. Lee, "Activity recognition based on semi-supervised learning," in *13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007).* IEEE Computer Society, 2007, pp. 469–475, doi: 10.1109/RTCSA.2007.17.

[39] A. Reiss and D. Stricker, "Personalized mobile physical activity recognition," in *Proceedings of the 2013 International Symposium on Wearable Computers.* ACM, 2013, pp. 25–28, doi: 10.1145/2493988.2494349.

[40] L. Gao, A. Bourke, and J. Nelson, "Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems," *Medical Engineering & Physics*, vol. 36, no. 6, pp. 779–785, 2014, doi: 10.1016/j.medengphy.2014.02.012.

[41] D. J. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowledge and Information Systems*, vol. 36, no. 3, pp. 537–556, 2013, doi: 10.1007/s10115-013-0665-3.

[42] A. Calatroni, D. Roggen, and G. Tröster, "Collection and curation of a large reference dataset for activity recognition," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE Computer Society, 2011, pp. 30–35, doi: 10.1109/ICSMC.2011.6083638.

[43] A. Reiss and D. Stricker, "Creating and benchmarking a new dataset for physical activity monitoring," in *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, 2012, pp. 40:1–40:8, doi: 10.1145/2413097.2413148.

[44] S. W. Loke, "Representing and reasoning with situations for context-aware pervasive computing: A logic programming perspective," *The Knowledge Engineering Review*, vol. 19, no. 3, pp. 213–233, 2004, doi: 10.1017/S0269888905000263.

[45] X. H. Wang, T. Gu, D. Q. Zhang, and H. K. Pung, "Ontology based context modeling and reasoning using OWL," in *Proceedings of Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*. IEEE Computer Society, 2004, pp. 18–22, doi: 10.1109/PERCOMW.2004.1276898.

[46] L. Chen and C. D. Nugent, "Ontology-based activity recognition in intelligent pervasive environments," *International Journal of Web Information Systems*, vol. 5, no. 4, pp. 410–430, 2009, doi: 10.1108/17440080911006199.

[47] R. Helaoui, D. Riboni, and H. Stuckenschmidt, "A probabilistic ontological framework for the recognition of multilevel human activities," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2013, pp. 345–354, doi: 10.1145/2493432.2493501.

[48] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos, "A survey on ambient intelligence in healthcare," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2470–2494, 2013, doi: 10.1109/JPROC.2013.2262913.

[49] J. Ye, S. Dobson, and S. McKeever, "Situation identification techniques in pervasive computing: A review," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 36–66, 2012, doi: 10.1016/j.pmcj.2011.01.004.

[50] J. Wen, J. Indulska, and M. Zhong, "Adaptive activity learning with dynamically available context," in *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE Computer Society, 2016, pp. 1–11, doi: 10.1109/PERCOM.2016.7456502.

[51] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes? Past, present, and future," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190–1203, 2012, doi: 10.1109/TSMCC.2012.2189204.

[52] N. Roy, A. Misra, and D. Cook, "Ambient and smartphone sensor assisted ADL recognition in multi-inhabitant smart environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 1, pp. 1–19, 2016, doi: 10.1007/s12652-015-0294-7.

[53] D. Riboni and M. Murtas, "Web mining & computer vision: New partners for object-based activity recognition," in *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE).* IEEE Computer Society, 2017, pp. 158–163, doi: 10.1109/WETICE.2017.38.

[54] J. Andreu, R. D. Baruah, and P. Angelov, "Real time recognition of human activities from wearable sensors by evolving classifiers," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ).* IEEE Computer Society, 2011, pp. 2786–2793, doi: 10.1109/FUZZY.2011.6007595.

[55] Y. J. Hong, I. J. Kim, S. C. Ahn, and H. G. Kim, "Activity recognition using wearable sensors for elder care," in *2008 Second International Conference on Future Generation Communication and Networking.* IEEE Computer Society, 2008, pp. 302–305, doi: 10.1109/FGCN.2008.165.

[56] Y. Liu, L. Nie, L. Liu, and D. S. Rosenblum, "From action to activity: Sensor-based activity recognition," *Neurocomputing*, vol. 181, pp. 108–115, 2016, doi: 10.1016/j.neucom.2015.08.096.

[57] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor activity context detection for wearable computing," in *Ambient Intelligence: First European Symposium, EUSAI 2003, Proceedings.* Springer Berlin Heidelberg, 2003, pp. 220–232, doi: 10.1007/978-3-540-39863-9_17.

[58] P. H. Veltink, H. J. Bussmann, W. de Vries, W. J. Martens, and R. C. V. Lummel, "Detection of static and dynamic activities using uniaxial accelerometers," *IEEE Transactions on Rehabilitation Engineering*, vol. 4, no. 4, pp. 375–385, 1996, doi: 10.1109/86.547939.

[59] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002, doi: 10.1109/MCOM.2002.1024422.

[60] B. Crowell, *Light and Matter*, 2018th ed., B. Crowell, Ed. California: Fullerton, 2013, http://www.lightandmatter.com/lm/.

[61] J. B. Scarborough, *The Gyroscope – Theory and Applications.* Interscience Publishers Limited, 1958. ISBN 978-0343183110 Archive.org/details/gyroscopetheorya010740mbp.

[62] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Computing Research Repository - arXiv*, vol. abs/1704.06053, pp. 1–92, 2017, http://arxiv.org/abs/1704.06053.

[63] J. McCann and D. Bryson, Eds., *Smart clothes and wearable technology.* Woodhead Publishing, 2009. ISBN 978-1-84569-357-2

[64] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators.* CRC press, 2010, vol. 39. ISBN 978-1-4398-2109-1

[65] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems 30.* Curran Associates, Inc., 2017, pp. 3146–3154.

[66] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys*, vol. 51, no. 5, pp. 92:1–92:36, 2018, doi: 10.1145/3234150.

[67] I. Rish, "An empirical study of the naive Bayes classifier," in *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.* IBM New York, 2001, pp. 41–46.

[68] P. M. Lee, *Bayesian statistics: An introduction.* John Wiley & Sons, 2012. ISBN 978-1118332573

[69] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967, doi: 10.1109/TIT.1967.1053964.

[70] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998, doi: 10.1109/5254.708428.

[71] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002, doi: 10.1109/72.991427.

[72] R. J. Schalkoff, *Artificial Neural Networks.* McGraw-Hill New York, 1997. ISBN 978-0070571181 OCoLC: 645853765.

[73] R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 93:1–93:42, 2018, doi: 10.1145/3236009.

[74] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986, doi: 10.1023/A:1022643204877.

[75] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the Gini index and information gain criteria," *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 77–93, 2004, doi: 10.1023/B:AMAI.0000018580.96245.c6.

[76] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[77] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998, doi: 10.1109/34.709601.

[78] N. C. Oza, "Online bagging and boosting," in *2005 IEEE International Conference on Systems, Man and Cybernetics.* IEEE Computer Society, 2005, pp. 2340–2345, doi: 10.1109/ICSMC.2005.1571498.

[79] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996, doi: 10.1007/BF00058655.

[80] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops).* IEEE Computer Society, 2009, pp. 1393–1400, doi: 10.1109/ICCVW.2009.5457447.

[81] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications*, 2nd ed. Cambridge University Press, 2010. ISBN 978-0521150118

[82] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler, "OWL 2: The next step for OWL," *Journal of Web Semantics*, vol. 6, no. 4, pp. 309–322, 2008, doi: 10.1016/j.websem.2008.05.001.

[83] J. Bock, P. Haase, Q. Ji, and R. Volz, "Benchmarking OWL reasoners," in *Workshop on Advancing Reasoning on the Web: Scalability and Commonsense (ARea2008).* CEUR-WS.org, 2008, pp. 1–15.

[84] L. De Raedt, A. Kimmig, and H. Toivonen, "Problog: A probabilistic prolog and its application in link discovery," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence.* Morgan Kaufmann Publishers Inc., 2007, pp. 2468–2473.

[85] M. Bruynooghe, T. Mantadelis, A. Kimmig, B. Gutmann, J. Vennekens, G. Janssens, and L. De Raedt, "Problog technology for inference in a probabilistic first order logic," in *Frontiers in Artificial Intelligence and Applications.* IOS Press, 2010, pp. 719–724, doi: 10.3233/978-1-60750-606-5-719.

[86] J. Noessner, M. Niepert, and H. Stuckenschmidt, “RockIt: Exploiting parallelism and symmetry for map inference in statistical relational models,” in *Proceedings of the 16th AAAI Conference on Statistical Relational Artificial Intelligence.* AAAI Press, 2013, pp. 37–42.

[87] S. Riedel, “Improving the accuracy and efficiency of map inference for Markov logic,” in *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, UAI 2008.* AUAI Press, 2008, pp. 468–475.

[88] F. Niu, C. Ré, A. Doan, and J. Shavlik, “Tuffy: Scaling up statistical inference in Markov logic networks using an RDBMS,” *Proceedings of the VLDB Endowment*, vol. 4, no. 6, pp. 373–384, 2011, doi: 10.14778/1978665.1978669.

[89] J. Huber, C. Meilicke, and H. Stuckenschmidt, “Applying Markov logic for debugging probabilistic temporal knowledge bases,” in *AKBC 2014: 4th Workshop on Automated Knowledge Base Construction AKBC 2014 at NIPS 2014 in Montreal, Canada, December 13, 2014.* ACM, 2014, pp. 1–6.

[90] M. Richardson and P. Domingos, “Markov logic networks,” *Machine learning*, vol. 62, no. 1, pp. 107–136, 2006, doi: 10.1007/s10994-006-5833-1.

[91] J. Pearl, *Probabilistic reasoning in intelligent systems: Networks of plausible inference*, 2nd ed. Morgan Kaufmann, 2014. ISBN 978-1558604797

[92] H. Rue and L. Held, *Gaussian Markov random fields: Theory and applications.* CRC Press, 2005. ISBN 978-0-203-49202-4

[93] R. M. Smullyan, *First-order logic.* Springer Science & Business Media, 1968, vol. 43. ISBN 978-3-642-86718-7

[94] M. Chekol, J. Huber, C. Meilicke, and H. Stuckenschmidt, “Markov logic networks with numerical constraints,” in *22st European Conference on Artificial Intelligence (ECAI2016).* IOS Press, 2016, pp. 1017–1025, doi: 10.3233/978-1-61499-672-9-1017.

[95] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, and D. Howard, “A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data,” *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 3, pp. 871–879, 2009, doi: 10.1109/TBME.2008.2006190.

[96] D. Maguire and R. Frisby, “Comparison of feature classification algorithm for activity recognition based on accelerometer and heart rate data,” in *9th. IT & T Conference*, 2009, p. 11.

[97] M. Zhang and A. A. Sawchuk, “A feature selection-based framework for human activity recognition using wearable multimodal sensors,” in *Proceedings of the 6th*

International Conference on Body Area Networks, ser. BodyNets '11. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, pp. 92–98. ISBN 978-1-936968-29-9

[98] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, p. 6474, 2014, doi: 10.3390/s140406474.

[99] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, "Dynamic sensor data segmentation for real-time knowledge-driven activity recognition," *Pervasive and Mobile Computing*, vol. 10, no. Part B, pp. 155–172, 2014, doi: 10.1016/j.pmcj.2012.11.004.

[100] A. Vahdatpour, N. Amini, and M. Sarrafzadeh, "Toward unsupervised activity discovery using multi-dimensional motif detection in time series," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI).* Morgan Kaufmann Publishers Inc., 2009, pp. 1261–1266.

[101] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010, doi: 10.3390/s100201154.

[102] A. M. Khan, A. Tufail, A. M. Khattak, and T. H. Laine, "Activity recognition on smartphones via sensor-fusion and KDA-based SVMs," *International Journal of Distributed Sensor Networks*, vol. 10, no. 5, pp. 1–15, 2014, doi: 10.1155/2014/503291.

[103] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster, "Where am I: Recognizing on-body positions of wearable sensors," in *International Symposium on Location- and Context-Awareness.* Springer Berlin Heidelberg, 2005, pp. 264–275, doi: 10.1007/11426646_25.

[104] S. Thiemjarus, "A device-orientation independent method for activity recognition," in *2010 International Conference on Body Sensor Networks.* IEEE Computer Society, 2010, pp. 19–23, doi: 10.1109/BSN.2010.55.

[105] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *Ubiquitous Intelligence and Computing: 7th International Conference, UIC 2010, Xi'an, China, October 26-29, 2010. Proceedings.* Springer Berlin Heidelberg, 2010, pp. 548–562, doi: 10.1007/978-3-642-16355-5_42.

[106] K. Altun and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," in *Human Behavior Understanding: First International Workshop, HBU 2010, Istanbul, Turkey, August 22, 2010. Proceedings.* Springer Berlin Heidelberg, 2010, pp. 38–51, doi: 10.1007/978-3-642-14715-9_5.

[107] D. De, P. Bharti, S. K. Das, and S. Chellappan, “Multimodal wearable sensing for fine-grained activity recognition in healthcare,” *IEEE Internet Computing*, vol. 19, no. 5, pp. 26–35, 2015, doi: 10.1109/MIC.2015.72.

[108] O. D. Lara and M. A. Labrador, “A survey on human activity recognition using wearable sensors,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013, doi: 10.1109/SURV.2012.110112.00192.

[109] B. Kolosnjaji and C. Eckert, *Neural Network-Based User-Independent Physical Activity Recognition for Mobile Devices.* Springer International Publishing, 2015, pp. 378–386. ISBN 978-3-319-24834-9

[110] Y. Nam, S. Rho, and C. Lee, “Physical activity recognition using multiple sensors embedded in a wearable device,” *ACM Trans. Embed. Comput. Syst.*, vol. 12, no. 2, pp. 26:1–26:14, 2013, doi: 10.1145/2423636.2423644.

[111] P. Casale, O. Pujol, and P. Radeva, “Human activity recognition from accelerometer data using a wearable device,” in *Pattern Recognition and Image Analysis.* Springer Berlin Heidelberg, 2011, pp. 289–296, doi: 10.1007/978-3-642-21257-4_36. ISBN 978-3-642-21257-4

[112] A. Mannini, S. S. Intille, M. Rosenberger, A. M. Sabatini, and W. Haskell, “Activity recognition using a single accelerometer placed at the wrist or ankle,” *Medicine and science in sports and exercise*, vol. 45, no. 11, pp. 2193–2203, 2013, doi: 10.1249/MSS.0b013e31829736d6.

[113] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, “Fusion of smartphone motion sensors for physical activity recognition,” *Sensors*, vol. 14, no. 6, pp. 10 146–10 176, 2014, doi: 10.3390/s140610146.

[114] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, “Sensor-based activity recognition,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, 2012, doi: 10.1109/TSMCC.2012.2198883.

[115] X. Su, H. Tong, and P. Ji, “Activity recognition with smartphone sensors,” *Tsinghua Science and Technology*, vol. 19, no. 3, pp. 235–249, 2014, doi: 10.1109/TST.2014.6838194.

[116] A. Henpraserttae, S. Thiemjarus, and S. Marukatat, “Accurate activity recognition using a mobile phone regardless of device orientation and location,” in *2011 International Conference on Body Sensor Networks (BSN).* IEEE Computer Society, 2011, pp. 41–46, doi: 10.1109/BSN.2011.8.

[117] H. Martín, A. M. Bernardos, J. Iglesias, and J. R. Casar, "Activity logging using lightweight classification techniques in mobile devices," *Personal and Ubiquitous Computing*, vol. 17, no. 4, pp. 675–695, 2013, doi: 10.1007/s00779-012-0515-4.

[118] A. Vahdatpour, N. Amini, and M. Sarrafzadeh, "On-body device localization for health and medical monitoring applications," in *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE Computer Society, 2011, pp. 37–44, doi: 10.1109/PERCOM.2011.5767593.

[119] E. Miluzzo, M. Papandrea, N. D. Lane, H. Lu, and A. T. Campbell, "Pocket, bag, hand, etc. - Automatically detecting phone context through discovery," in *First International Workshop on Sensing for App Phones (PhoneSense 2010)*, 2010, pp. 21–25.

[120] J.-G. Park, A. Patel, D. Curtis, S. Teller, and J. Ledlie, "Online pose classification and walking speed estimation using handheld devices," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing.* ACM, 2012, pp. 113–122, doi: 10.1145/2370216.2370235.

[121] C. T. Cornelius and D. F. Kotz, "Recognizing whether sensors are on the same body," *Pervasive and Mobile Computing*, vol. 8, no. 6, pp. 822–836, 2012, doi: 10.1016/j.pmcj.2012.06.005.

[122] G. M. Weiss and J. W. Lockhart, "The impact of personalization on smartphone-based activity recognition," in *AAAI Workshop on Activity Context Representation: Techniques and Languages.* AAAI Press, 2012, pp. 98–104.

[123] T. Maekawa and S. Watanabe, "Unsupervised activity recognition with user's physical characteristics data," in *2011 15th Annual International Symposium on Wearable Computers.* IEEE Computer Society, 2011, pp. 89–96, doi: 10.1109/ISWC.2011.24.

[124] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu, "Cross-people mobile-phone based activity recognition," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Three.* AAAI Press, 2011, pp. 2545–2550, doi: 10.5591/978-1-57735-516-8/IJCAI11-423.

[125] W.-Y. Deng, Q.-H. Zheng, and Z.-M. Wang, "Cross-person activity recognition using reduced kernel extreme learning machine," *Neural Networks*, vol. 53, pp. 1–7, 2014, doi: 10.1016/j.neunet.2014.01.008.

[126] M. Stikic, K. V. Laerhoven, and B. Schiele, "Exploring semi-supervised and active learning for activity recognition," in *2008 12th IEEE International Symposium on Wearable Computers.* IEEE Computer Society, 2008, pp. 81–88, doi: 10.1109/ISWC.2008.4911590.

[127] R. Liu, T. Chen, and L. Huang, "Research on human activity recognition based on active learning," in *2010 International Conference on Machine Learning and Cybernetics*. IEEE Computer Society, 2010, pp. 285–290, doi: 10.1109/ICMLC.2010.5581050.

[128] L. Yao, F. Nie, Q. Z. Sheng, T. Gu, X. Li, and S. Wang, "Learning from less for better: Semi-supervised activity recognition via shared structure discovery," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 13–24, doi: 10.1145/2971648.2971701.

[129] H. Alemdar, T. L. M. van Kasteren, and C. Ersoy, "Using active learning to allow activity recognition on a large scale," in *Ambient Intelligence: Second International Joint Conference on AmI 2011, Amsterdam, The Netherlands, November 16-18, 2011. Proceedings*. Springer Berlin Heidelberg, 2011, pp. 105–114, doi: 10.1007/978-3-642-25167-2_12.

[130] J. Pärkkä, L. Cluitmans, and M. Ermes, "Personalization algorithm for real-time activity recognition using PDA, wireless motion bands, and binary decision tree," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 5, pp. 1211–1215, 2010, doi: 10.1109/TITB.2010.2055060.

[131] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "StreamAR: Incremental and active learning with evolving sensory data for activity recognition," in *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*. IEEE Computer Society, 2012, pp. 1163–1170, doi: 10.1109/ICTAI.2012.169.

[132] Z. Wang, M. Jiang, Y. Hu, and H. Li, "An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 4, pp. 691–699, 2012, doi: 10.1109/TITB.2012.2196440.

[133] Z. S. Abdallah, M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Adaptive mobile activity recognition system with evolving data streams," *Neurocomputing*, vol. 150, no. Part A, pp. 304–317, 2015, doi: 10.1016/j.neucom.2014.09.074.

[134] Y. A. Jeroudi, M. A. Ali, M. Latief, and R. Akmeliawati, "Online sequential extreme learning machine algorithm based human activity recognition using inertial data," in *2015 10th Asian Control Conference (ASCC)*. IEEE Computer Society, 2015, pp. 1–6, doi: 10.1109/ASCC.2015.7244597.

[135] Q. V. Vo, M. T. Hoang, and D. Choi, "Personalization in mobile activity recognition system using k-medoids clustering algorithm," *International Journal of Distributed Sensor Networks*, vol. 9, no. 7, pp. 1–12, 2013, doi: 10.1155/2013/315841.

[136] J. Andreu and P. Angelov, "An evolving machine learning method for human activity recognition systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 4, no. 2, pp. 195–206, 2013, doi: 10.1007/s12652-011-0068-9.

[137] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A hybrid discriminative/generative approach for modeling human activities," in *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2005, pp. 766–772.

[138] P. Lukowicz, J. A. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, and T. Starner, "Recognizing workshop activity using body worn microphones and accelerometers," in *Pervasive Computing: Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria, April 21-23, 2004. Proceedings.* Springer Berlin Heidelberg, 2004, pp. 18–32, doi: 10.1007/978-3-540-24646-6_2.

[139] T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu, "epSICAR: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition," in *Proceedings of the Seventh Annual IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE Computer Society, 2009, pp. 1–9, doi: 10.1109/PERCOM.2009.4912776.

[140] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. del R. Millán, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proceedings of the Seventh International Conference on Networked Sensing Systems.* IEEE Computer Society, 2010, pp. 233–240, doi: 10.1109/INSS.2010.5573462.

[141] D. H. Hu and Q. Yang, "Transfer learning for activity recognition via sensor mapping," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Three.* AAAI Press, 2011, pp. 1962–1967, doi: 10.5591/978-1-57735-516-8/IJCAI11-328.

[142] D. Riboni and C. Bettini, "OWL 2 modeling and reasoning with complex human activities," *Pervasive and Mobile Computing*, vol. 7, no. 3, pp. 379–395, 2011, doi: 10.1016/j.pmcj.2011.02.001.

[143] J. Ye and G. Stevenson, "Semantics-driven multi-user concurrent activity recognition," in *Ambient Intelligence: 4th International Joint Conference, AmI 2013, Dublin, Ireland, December 3-5, 2013. Proceedings.* Springer International Publishing, 2013, pp. 204–219, doi: 10.1007/978-3-319-03647-2_15.

[144] G. Meditskos, E. Kontopoulos, and I. Kompatsiaris, "Knowledge-driven activity recognition and segmentation using context connections," in *The Semantic Web – ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy,*

October 19-23, 2014. Proceedings, Part II. Springer International Publishing, 2014, pp. 260–275, doi: 10.1007/978-3-319-11915-1_17.

[145] D. Riboni and C. Bettini, "COSAR: Hybrid reasoning for context-aware activity recognition," *Personal and Ubiquitous Computing*, vol. 15, no. 3, pp. 271–289, 2011, doi: 10.1007/s00779-010-0331-7.

[146] G. Meditskos, E. Kontopoulos, and I. Kompatsiaris, "ReDef: Context-aware recognition of interleaved activities using OWL 2 and defeasible reasoning," in *Joint Proceedings of SSN-TC and OrdRing 2015.* CEUR-WS.org, 2015, pp. 31–42.

[147] C. W. Geib and R. P. Goldman, "A probabilistic plan recognition algorithm based on plan tree grammars," *Artificial Intelligence*, vol. 173, no. 11, pp. 1101–1132, 2009, doi: 10.1016/j.artint.2009.01.003.

[148] M. Perkowitz, M. Philipose, K. P. Fishkin, and D. J. Patterson, "Mining models of human activities from the web," in *Proceedings of the 13th International Conference on World Wide Web.* ACM, 2004, pp. 573–582, doi: 10.1145/988672.988750.

[149] D. Wyatt, M. Philipose, and T. Choudhury, "Unsupervised activity recognition using automatically mined common sense," in *Proceedings of the 20th National Conference on Artificial Intelligence.* AAAI Press, 2005, pp. 21–27.

[150] E. M. Tapia, T. Choudhury, and M. Philipose, "Building reliable activity models using hierarchical shrinkage and mined ontology," in *Proceedings of International Conference on Pervasive Computing.* Springer Berlin Heidelberg, 2006, pp. 17–32, doi: 10.1007/11748625_2.

[151] P. Palmes, H. K. Pung, T. Gu, W. Xue, and S. Chen, "Object relevance weight pattern mining for activity recognition and segmentation," *Pervasive and Mobile Computing*, vol. 6, no. 1, pp. 43–57, 2010, doi: 10.1016/j.pmcj.2009.10.004.

[152] P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe, "Discovering activities to recognize and track in a smart environment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 4, pp. 527–539, 2011, doi: 10.1109/TKDE.2010.148.

[153] J. Ye, G. Stevenson, and S. Dobson, "USMART: An unsupervised semantic mining activity recognition technique," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 4, no. 4, pp. 16:1–16:27, 2014, doi: 10.1145/2662870.

[154] J. Yin, Q. Yang, D. Shen, and Z.-N. Li, "Activity recognition via user-trace segmentation," *ACM Transactions on Sensor Networks*, vol. 4, no. 4, pp. 19:1–19:34, 2008, doi: 10.1145/1387663.1387665.

[155] J. Wan, M. J. O'Grady, and G. M. O'Hare, "Dynamic sensor event segmentation for real-time activity recognition in a smart home context," *Personal and Ubiquitous Computing*, vol. 19, no. 2, pp. 287–301, 2015, doi: 10.1007/s00779-014-0824-x.

[156] S. Aminikhanghahi and D. J. Cook, "Using change point detection to automate daily activity segmentation," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE Computer Society, 2017, pp. 262–267, doi: 10.1109/PERCOMW.2017.7917569.

[157] D. Triboan, L. Chen, F. Chen, and Z. Wang, "Semantic segmentation of real-time sensor data stream for complex activity recognition," *Personal and Ubiquitous Computing*, vol. 21, no. 3, pp. 411–425, 2017, doi: 10.1007/s00779-017-1005-5.

[158] Y. Ho, C. Lu, I. Chen, S. Huang, C. Wang, L. Fu *et al.*, "Active-learning assisted self-reconfigurable activity recognition in a dynamic environment," in *2009 IEEE International Conference on Robotics and Automation*. IEEE Computer Society, 2009, pp. 813–818, doi: 10.1109/ROBOT.2009.5152428.

[159] L. Zhao, G. Sukthankar, and R. Sukthankar, "Robust active learning using crowdsourced annotations for activity recognition," in *Proceedings of the 11th AAAI Conference on Human Computation*. AAAI Press, 2011, pp. 74–79.

[160] W. S. Lasecki, Y. C. Song, H. A. Kautz, and J. P. Bigham, "Real-time crowd labeling for deployable activity recognition," in *Proceedings of Computer Supported Cooperative Work (CSCW)*. ACM, 2013, pp. 1203–1212, doi: 10.1145/2441776.2441912.

[161] H. M. S. Hossain, M. A. A. H. Khan, and N. Roy, "Active learning enabled activity recognition," *Pervasive and Mobile Computing*, vol. 38, no. Part 2, pp. 312–330, 2017, doi: 10.1016/j.pmcj.2016.08.017.

[162] E. Hoque and J. A. Stankovic, "AALO: Activity recognition in smart homes using active learning in the presence of overlapped activities," in *Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. IEEE Computer Society, 2012, pp. 139–146, doi: 10.4108/icst.pervasivehealth.2012.248600.

[163] D. Roggen, K. Forster, A. Calatroni, T. Holleczek, Y. Fang, G. Troster, A. Ferscha, C. Holzmann, A. Riener, P. Lukowicz, G. Pirkl, D. Bannach, K. Kunze, R. Chavarriaga, and J. d. R. Millan, "OPPORTUNITY: Towards opportunistic activity and context recognition systems," in *2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops (WoWMoM)*. IEEE Computer Society, 2009, pp. 1–6, doi: 10.1109/WOWMOM.2009.5282442.

[164] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural clas-

sifiers," *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2213–2220, 2008, doi: 10.1016/j.patrec.2008.08.002.

[165] S. Shen, H. Wang, and R. Roy Choudhury, "I am a smartwatch and I can track my user's arm," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2016, pp. 85–96, doi: 10.1145/2906388.2906407.

[166] M. Shoaib, J. Scholten, and P. Havinga, "Towards physical activity recognition using smartphone sensors," in *10th IEEE International Conference on Ubiquitous Intelligence and Computing, UIC 2013*. IEEE Computer Society, 2013, pp. 80–87, doi: 10.1109/UIC-ATC.2013.43.

[167] F. B. Hu, T. Y. Li, G. A. Colditz, W. C. Willett, and J. E. Manson, "Television watching and other sedentary behaviors in relation to risk of obesity and type 2 diabetes mellitus in women," *JAMA*, vol. 289, no. 14, pp. 1785–1791, 2003, doi: 10.1001/jama.289.14.1785.

[168] S. Krishnan, L. Rosenberg, and J. R. Palmer, "Physical activity and television watching in relation to risk of type 2 diabetes: The black women's health study," *American Journal of Epidemiology*, vol. 169, no. 4, pp. 428–434, 2008, doi: 10.1093/aje/kwn344.

[169] M. M. Anthimopoulos, L. Gianola, L. Scarnato, P. Diem, and S. G. Mougiakakou, "A food recognition system for diabetic patients based on an optimized bag-of-features model," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1261–1271, 2014, doi: 10.1109/JBHI.2014.2308928.

[170] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 119–128, 2006, doi: 10.1109/TITB.2005.856863.

[171] J. Ryder, B. Longstaff, S. Reddy, and D. Estrin, "Ambulation: A tool for monitoring mobility patterns over time using mobile phones," in *2009 International Conference on Computational Science and Engineering*, vol. 4, 2009, pp. 927–931, doi: 10.1109/CSE.2009.312.

[172] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, pp. 13:1–13:27, 2010, doi: 10.1145/1689239.1689243.

[173] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, "Cooperative transit tracking using smart-phones," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 85–98, doi: 10.1145/1869983.1869993.

[174] K. Muralidharan, A. J. Khan, A. Misra, R. K. Balan, and S. Agarwal, "Barometric phone sensors: More hype than hope!" in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications.* ACM, 2014, pp. 12:1–12:6, doi: 10.1145/2565585.2565596.

[175] M. Han, L. T. Vinh, Y.-K. Lee, and S. Lee, "Comprehensive context recognizer based on multimodal sensors in a smartphone," *Sensors*, vol. 12, no. 9, pp. 12 588–12 605, 2012, doi: 10.3390/s120912588.

[176] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking.* ACM, 2015, pp. 65–76, doi: 10.1145/2789168.2790093.

[177] L. Diambra, J. de Figueiredo, and C. Malta, "Epileptic activity recognition in EEG recording," *Physica A: Statistical Mechanics and its Applications*, vol. 273, no. 3, pp. 495–505, 1999, doi: 10.1016/S0378-4371(99)00368-4.

[178] X. Zhang, L. Yao, D. Zhang, X. Wang, Q. Z. Sheng, and T. Gu, "Multi-person brain activity recognition via comprehensive EEG signal analysis," *Computing Research Repository - arXiv*, vol. abs/1709.09077, pp. 1–10, 2017, https://arxiv.org/abs/1709.09077.

[179] K. Zhan, F. Ramos, and S. Faux, "Activity recognition from a wearable camera," in *2012 12th International Conference on Control Automation Robotics Vision (ICARCV).* IEEE Computer Society, 2012, pp. 365–370, doi: 10.1109/ICARCV.2012.6485186.

[180] G. Ballin, M. Munaro, and E. Menegatti, "Human action recognition from RGB-D frames based on real-time 3D optical flow estimation," in *Biologically Inspired Cognitive Architectures.* Springer Berlin Heidelberg, 2013, pp. 65–74, doi: 10.1007/978-3-642-34274-5_17.

[181] J. Aggarwal and L. Xia, "Human activity recognition from 3D data: A review," *Pattern Recognition Letters*, vol. 48, pp. 70–80, 2014, doi: 10.1016/j.patrec.2014.04.011.

[182] A. Betancourt, P. Morerio, C. S. Regazzoni, and M. Rauterberg, "The evolution of first person vision methods: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 744–760, 2015, doi: 10.1109/TCSVT.2015.2409731.

[183] P.-Y. Hsu, W.-F. Cheng, P.-J. Hsieh, Y.-L. Lin, and W. H. Hsu, "Real-time instant event detection in egocentric videos by leveraging sensor-based motion context," in *Proceedings of the 23rd ACM International Conference on Multimedia.* ACM, 2015, pp. 1275–1278, doi: 10.1145/2733373.2806404.

[184] T. Miyanishi, J. ichiro Hirayama, Q. Kong, T. Maekawa, H. Moriya, and T. Suyama, "Egocentric video search via physical interactions," in *13th AAAI Conference on Artificial Intelligence*, 2016, pp. 330–337.

[185] L. Ryoo, M. S.and Matthies, "First-person activity recognition: Feature, temporal structure, and prediction," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 307–328, 2016, doi: 10.1007/s11263-015-0847-4.

[186] G. Singla, D. J. Cook, and M. Schmitter-Edgecombe, "Tracking activities in complex settings using smart environment technologies," *International Journal of Biosciences, Psychiatry, and Technology*, vol. 1, no. 1, pp. 25–35, 2009.

[187] B. Reisberg, S. Finkel, J. Overall, N. Schmidt-Gollas, S. Kanowski, H. Lehfeld, F. Hulla, S. G. Sclan, H.-U. Wilms, K. Heininger, and et al., "The Alzheimer's disease activities of daily living international scale (ADL-IS)," *International Psychogeriatrics*, vol. 13, no. 2, pp. 163–181, 2001, doi: 10.1017/S1041610201007566.

[188] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and V. Bulgari, "From lab to life: Fine-grained behavior monitoring in the elderly's home," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops).* IEEE Computer Society, 2015, pp. 342–347, doi: 10.1109/PERCOMW.2015.7134060.

[189] R. Helaoui, M. Niepert, and H. Stuckenschmidt, "Recognizing interleaved and concurrent activities: A statistical-relational approach," in *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE Computer Society, 2011, pp. 1–9, doi: 10.1109/PERCOM.2011.5767586.

[190] P. Chahuara, A. Fleury, F. Portet, and M. Vacher, "Using Markov logic network for on-line activity recognition from non-visual home automation sensors," *Ambient Intelligence: Third International Joint Conference, AmI 2012, Pisa, Italy, November 13-15, 2012. Proceedings*, pp. 177–192, 2012, doi: 10.1007/978-3-642-34898-3_12.

[191] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui, "Fine-grained recognition of abnormal behaviors for early detection of mild cognitive impairment," in *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE Computer Society, 2015, pp. 149–154, doi: 10.1109/PERCOM.2015.7146521.

[192] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and Mobile Computing*, vol. 10, no. Part B, pp. 138–154, 2014, doi: 10.1016/j.pmcj.2012.07.003.

[193] P. Voigt and A. Von dem Bussche, *The EU General Data Protection Regulation (GDPR).* Springer, 2017, vol. 18. ISBN 978-3-319-57958-0

[194] P. Rashidi and D. J. Cook, "Keeping the resident in the loop: Adapting the smart home to the user," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 39, no. 5, pp. 949–959, 2009, doi: 10.1109/TSMCA.2009.2025137.

[195] A. B. Karami, A. Fleury, J. Boonaert, and S. Lecoeuche, "User in the loop: Adaptive smart homes exploiting user feedback – State of the art and future directions," *Information*, vol. 7, no. 2, pp. 1–18, 2016, doi: 10.3390/info7020035.

[196] A. Aztiria, A. Izaguirre, R. Basagoiti, and J. C. Augusto, "Learning about preferences and common behaviours of the user in an intelligent environment," *Behaviour Monitoring and Interpretation – BMI*, vol. 3, pp. 289–315, 2009, doi: 10.3233/978-1-60750-048-3-289.

[197] J. Yin, M. Fang, G. Mokhtari, and Q. Zhang, "Multi-resident location tracking in smart home through non-wearable unobtrusive sensors," in *International Conference on Smart Homes and Health Telematics (ICOST 2016).* Springer International Publishing, 2016, pp. 3–13, doi: 10.1007/978-3-319-39601-9_1.

[198] A. Benmansour, A. Bouchachia, and M. Feham, "Modeling interaction in multi-resident activities," *Neurocomputing*, vol. 230, pp. 133–142, 2017, doi: 10.1016/j.neucom.2016.05.110.

[199] H. Alemdar and C. Ersoy, "Multi-resident activity tracking and recognition in smart environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 4, pp. 513–529, 2017, doi: 10.1007/s12652-016-0440-x.

[200] A. Alhamoud, V. Muradi, D. Böhnstedt, and R. Steinmetz, "Activity recognition in multi-user environments using techniques of multi-label classification," in *Proceedings of the 6th International Conference on the Internet of Things.* ACM, 2016, pp. 15–23, doi: 10.1145/2991561.2991563.

[201] M. A. U. Alam, N. Roy, A. Misra, and J. Taylor, "CACE: Exploiting behavioral interactions for improved activity recognition in multi-inhabitant smart homes," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS).* IEEE Computer Society, 2016, pp. 539–548, doi: 10.1109/ICDCS.2016.61.

[202] R. Mohamed, T. Perumal, M. N. Sulaiman, N. Mustapha, and M. N. S. Zainudin, "Multi label classification on multi resident in smart home using classifier chains," *Advanced Science Letters*, vol. 24, no. 2, pp. 1316–1319, 2018, doi: 10.1166/asl.2018.10740.

[203] G. Mokhtari, A. Anvari-Moghaddam, Q. Zhang, and M. Karunanithi, "Multi-residential activity labelling in smart homes with wearable tags using BLE technology," *Sensors*, vol. 18, no. 3, pp. 1–16, 2018, doi: 10.3390/s18030908.

[204] A. A. Atayero and O. Feyisetan, “Security issues in cloud computing: The potentials of homomorphic encryption,” *Journal of Emerging Trends in Computing and Information Sciences*, vol. 2, no. 10, pp. 546–552, 2011.

[205] Y. Lindell and B. Pinkas, “Secure multiparty computation for privacy-preserving data mining,” *Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 59–98, 2009.

[206] C. Roussey, F. Pinet, M. A. Kang, and O. Corcho, *An Introduction to Ontologies and Ontology Engineering.* Springer London, 2011, pp. 9–38. ISBN 978-0-85729-724-2

[207] A. Diete, T. Sztyler, L. Weiland, and H. Stuckenschmidt, “Improving motion-based activity recognition with ego-centric vision,” in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops).* IEEE Computer Society, 2018, pp. 653–656, doi: 10.1109/PERCOMW.2018.8480334.

[208] Y. Wang, S. Cang, and H. Yu, “A data fusion based hybrid sensory system for older people’s daily activity and daily routine recognition,” *IEEE Sensors Journal*, vol. 18, no. 16, pp. 6874–6888, 2018, doi: 10.1109/JSEN.2018.2833745.

[209] A. Diete, T. Sztyler, L. Weiland, and H. Stuckenschmidt, “Recognizing grabbing actions from inertial and video sensor data in a warehouse scenario,” *Procedia Computer Science*, vol. 110, pp. 16–23, 2017, doi: 10.1016/j.procs.2017.06.071.

[210] T. Sztyler, J. Völker, J. Carmona, O. Meier, and H. Stuckenschmidt, “Discovery of personal processes from labeled sensor data - An application of process mining to personalized health care,” in *Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data, ATAED.* CEUR-WS.org, 2015, pp. 31–46. ISSN 1613-0073

[211] T. Sztyler, J. Carmona, J. Völker, and H. Stuckenschmidt, *Self-Tracking Reloaded: Applying Process Mining to Personalized Health Care from Labeled Sensor Data.* Springer-Verlag Berlin Heidelberg, 2016, vol. 9930, pp. 160–180.

[212] T. Sztyler, “Towards real world activity recognition from wearable devices,” in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops).* IEEE Computer Society, 2017, pp. 97–98, doi: 10.1109/PERCOMW.2017.7917535.