IMPROVED TRAINING APPROACHES FOR EMBEDDED LEARNING WITH HETEROGENEOUS SENSOR DATA

Doctoral Thesis

Submitted to the School of Science and Technology of the University of Siegen

for obtaining the doctoral degree

Doctor of Engineering (Dr.-Ing.)

by **Alexander Hölzemann**

Siegen, 12th December 2023

Created with LaTeX on Wednesday 5th June, 2024 09:40(CEST)

Supervisor and first reviewer
Prof. Dr. Kristof Van Laerhoven
University of Siegen, Germany

Second reviewer
Prof. Dr. Qin Lv
University of Colorado Boulder, USA

Day of defense
6th May 2024

Declaration of Authorship

I, Alexander Hölzemann, hereby declare that this doctoral thesis, entitled *Improved Training Approaches for Embedded Learning with Heterogeneous Sensor Data*, and the work presented within have been carried out by myself and that I have written this thesis independently. I confirm that this work has not already been submitted in whole or in part to satisfy any other degree. Where work from other authors has been included in this thesis, it has been properly cited and referenced. Those who have provided resources or data that I have cited and referenced have been identified in the relevant sections of the text.

I have also included a comprehensive list of the relevant references used in the production of this thesis within the References or Bibliography section.

Acknowledgments

I would like to express my deepest gratitude to my parents for their endless love, support, and encouragement throughout my academic journey. Their unwavering belief in me gave me the strength to pursue my doctorate.

I owe immense gratitude to my friends for their unwavering emotional, logistical, and social support. In particular, I wish to extend my heartfelt thanks to my basketball teammates from the TuS Fellinghausen 1920 e.V. for assisting me in recording the Hang-Time HAR dataset. Without their compassion and willingness to volunteer their time, I could not have completed this undertaking.

My sincerest appreciation to Prof. Van Laerhoven for his guidance and for nurturing my interest in machine learning for sensor data analysis. His supervision during my Master's thesis sparked my passion for this research area. My fascination with machine and deep learning was born during my Master's thesis in a steel-producing company under Prof. Van Laerhoven's direction. I applied machine learning to correlate manufacturing sensor data with product quality. This experience led me to join Prof. Van Laerhoven's Ubiquitous Computing group for my Ph.D. studies. At an early stage, my advisor and I chose to focus my thesis on enhancing activity recognition through machine learning, using sports and daily living as use cases. This decision marked the beginning of an intense, rewarding journey that advanced my research skills and defined my academic path.

I would love to express my gratitude to Prof. Dr. Qin Lv for serving as the second examiner on my thesis committee. Additionally, I would like to thank Prof. Dr. Roland Wiesmüller, chair of the board of examiners, and Prof. Dr. Madjid Fathi, associate chair, for their time and consideration of my thesis.

Finally, I must express my immense and profound gratitude to my partner and great love, Eldy Lazaro. Your insatiable curiosity and quest for understanding inspire me to continually grow and evolve. The unconditional love and unwavering support you provide create a secure foundation upon which I can build my life. You nurture my personal growth, challenging me to become the best version of myself, while simultaneously filling my world with radiant joy. You are the bedrock, the unshakable core, that everything else rests upon. Thank you for your presence in my life. I love you deeply.

Preface

Elements of the research comprising this thesis have been published in several peer-reviewed conference proceedings or journals as referenced. As this dissertation is presented in manuscript format, there is some unavoidable repetition in the introductory and methodological material across the component manuscripts. Efforts have been made to reduce redundancy where possible and reference relevant descriptions in other sections rather than reproducing content verbatim. Some duplication remains necessary to ensure that each manuscript constitutes a self-contained body of work.

I have been careful not to reproduce any copyrighted material. The content of this Ph.D. thesis was authored solely by myself. I did employ AI-driven tools, such as ChatGPT [158], Claude.ai [13], Grammarly [2] and DeepL [1], solely to enhance readability by utilizing these tools to rectify grammar, spelling, and maintain a scientific tone.

Abstract

The papers published as part of this doctoral thesis address significant challenges in the areas of annotation and synchronization of datasets, data-driven studies in the field of human-computer interaction, and machine learning related to multimodal activity data. Specifically, annotation workflows are examined for their robustness, an algorithm for improved synchronization of different acceleration data is presented, and potential data distortions in supervised studies involving human subjects are investigated within the context of a data-driven study. Advanced deep learning methods are employed for the analysis of human activities based on sensor data. Furthermore, an extensive dataset of motion data from basketball players has been released. Finally, the applicability of deep learning techniques, specifically *Transfer Learning* and *Data Augmentation*, to sensor data is explored.

The first research field explores real-world user studies for Human Activity Recognition (HAR) and introduces the Activate-System. This system enables ad-hoc data collection using a smartwatch and smartphone app. Furthermore, a presented user study evaluates and improves data collection and annotation methodologies. A 2-week study compares self-report diaries and in-situ annotation techniques, finding visualizing sensor data as time series improves recall accuracy. Additionally, I present an algorithm that synchronizes signals from multiple on-body sensors by exploiting cross-correlations of acceleration signals. Finally, the Hawthorne Effect is analyzed by collecting observed and unobserved data. This effect suggests that study participants alter their behavior once they are aware of being part of a study or under observation.

The second focus is recognizing activities in complex environments like sports, specifically basketball. A preliminary study shows the feasibility of detecting fine-grained basketball activities using a single wrist-worn inertial sensor. The Hang-Time HAR dataset, with data from 24 players, encompasses periodic, sporadic, and complex basketball movements, enabling comprehensive classification through deep learning.

The third contribution focuses on transfer learning and data augmentation for sensor-based activity recognition. Extensive experiments assess model transferability across sensor positions, modalities, and activity domains. Results reveal high variability depending on specific factors, such as body location, and deteriorating when source and target domains differ significantly. Data augmentation with Generative Adversarial Network (GAN) is also explored, comparing user-wise and fold-wise synthetic data generation. Expanding the dataset size by a factor of five improves the F1-Score by 11.0% for user-wise augmentation and 5.1% for fold-wise augmentation.

Contents

Chapter 1

Introduction

The integration of wearables into the everyday routines of many people is often driven by the desire to live a conscious and active life, or by the need for medical monitoring due to pre-existing conditions. Due to these circumstances and the resulting, high health-promoting potential, the technological development of wearables as well as the trialing of new types of devices and principles is still increasing and therefore the vision that such devices shall fade into the background and become an invisible part of everyone's life, formulated by Mark Weiser in 1991 [223], doesn't seem to be in far distance anymore.

Beyond everyday uses, activity tracking based on Inertial Measurement Units (IMUs) opens up numerous possibilities within more specialized domains.

A notable example is the collaborative effort between researchers at the University of Sussex and the car manufacturer Skoda, the Skoda-mini dataset [237]. A dataset that categorizes assembly line workers' tasks into distinct activity classes. Other companies utilize inertial measurement units in conjunction with video or indoor positioning systems, such as IndoTrack [119], which can then be leveraged to refine pose estimation and motion tracking capabilities. Enhanced pose estimation enables ergonomic analysis and informs workplace modifications to mitigate injury risk and enhance performance for laborers in physically strenuous occupations [121]. In sports science, IMUs embedded in wearable activity trackers are utilized to quantify biomechanics through motion capture and analysis of movement patterns and technique execution. Collecting data with IMUs enables the objective evaluation and profiling of an athlete's execution form, power, and efficiency across training exercises and competitive movements. Biomechanical feedback made possible by IMUs facilitates the systematic identification of strengths, weaknesses, and asymmetries in an athlete's mechanics. This evidence-based insight allows sports scientists and coaches to optimize training programs, correct poor technique, and provide tailored recommendations for performance enhancement and injury prevention [93]. Medical publications often focus on understanding diseases that cause motoric disorders, like Parkinson's, Diabetes, Spinal Muscular Atrophy, Huntington's disease, Amyotrophic Lateral Sclerosis, or sports injuries, and helping patients who suffer from these diseases by using devices

that incorporate IMUs. In addition to activity recognition and tracking applications, advancement in inertial measurement unit technology is significantly driven by demand from the military industry, as IMUs are a vital component in many weapon systems including missile and unmanned aerial vehicles navigation, and guidance systems. The defense industry fuels advancement in IMU technology to meet demands for lightweight, miniaturized, and highly precise inertial navigation and motion tracking capabilities.

Due to this demand from multiple domains, the unit market size for IMUs is forecasted to grow from 17.5 billion USD in 2021 to 24.5 USD in 2026 with an annual market growth of 7.21%, according to Global Market Estimates Inc. [77].

Section 1.1
Motivation

As a passionate basketball player since a young age, I was very eager about being able to combine my favorite hobby with my work as a computer scientist.

However, at a very early stage, I realized that most of the datasets available focus on activities of daily living, and sports are often just covered with very basic classes, see Section 4.2. Even though sports activities can be seen as the perfect playground to evaluate the limitations and opportunities for activity recognition - due to the naturally occurring mixture of periodic, sporadic, and complex activities - there was no dataset available that combines these characteristics and focuses on fine-grained activity recognition based on IMU sensor data. This finding inspired us to start with a feasibility study, see Chapter 4.1 or the corresponding publication [89], in order to test if fine-grained sports activities can be classified by using machine learning algorithms. The findings were promising and it motivated me to push the study further. At a later point during my Ph.D. - when the restrictions regarding the COVID-19 pandemic were lowered and the Activate System, see Section 3.1 and [92], was published, I decided to record and publish a comprehensive dataset with fine-grained basketball activities, including 24 subjects, and all categories of activities (periodic, sporadic, complex) recorded in both types on environments (controlled and uncontrolled). This study is presented in detail in Chapter 4.2 and [93]. This dataset has been recorded with one wrist-worn sensor. However, my initial plan was to include a second IMU in the form of an earbud called the eSense (Section 2.2), aiming to create a dataset with multiple types of data inputs.

It is common for HAR researchers to work with a multimodal input. This means, that the input data for machine or deep learning methodologies were recorded with multiple sensors installed on the same device or even multiple devices worn on different body locations with either a full (accelerometer, gyroscope, and magnetometer) or reduced IMU or other sensors like a Photoplethysmogram (PPG), air pressure or temperature sensors installed. However, using a multimodal approach generates additional hurdles, like the synchronization of independently operating sensor devices. Both devices run

with their own internal clock and it is difficult to synchronize them perfectly. This effect does generate an offset in the time-stamps of the signals, which needs to be adjusted before the data can be fed to a machine learning algorithm. This is still an ongoing research topic which many of the available datasets solved by adjusting the signals manually. However, this solution is time-consuming, error-prone, and labor-intensive. Chapter 3.3 as well as the publication [91] addresses this problem and presents a synchronization algorithm that is based on the cross-correlation between two on-body worn accelerometers.

Upon joining the ActiVAtE_Prevention project in 2021, the focus of my Ph.D. took a new direction, transitioning from sports activity recognition to the recording and annotation of long-term studies encompassing daily activities. This project monitored diabetes type 2 patients with the wearable smartwatch Bangle.js 1. In order to perform such a long-term study I needed to develop an operating system for the smartwatch and a smartphone app that was capable of downloading the activity data from the wearable device and uploading it to a database. This software stands as a foundational element for many of the papers presented in this thesis, thus assuming a critical role in shaping the conclusive results of this manuscript. Besides the recording of the dataset, working on this project led me to a deeper investigation of the effect of uncertain annotations for deep learning models and the usage of different annotation methods for studies conducted in-the-wild, presented in 3.2 and [96], where obtaining reliable ground truth is difficult. Moreover, I investigated the influence of the Hawthorne Effect on deep learning models, see Section 3.4. This phenomenon arises when study participants modify their behavior due to being under observation during experiments.

While the first two chapters explore user-driven annotation, behavior analysis, and basketball activity recognition specifically, this thesis provides also a novel investigation into two key deep learning techniques - Transfer Learning and Data Augmentation - for Human Activity Recognition using inertial measurement data. The concept of experimenting with Transfer Learning for the purpose of personalizing pre-trained classifiers originated from the basketball feasibility study, Section 4.1. The intention was to leverage the advantages of this approach to expedite training time or enhance the overall classifier performance by transferring the model weights from one player to another. Transfer Learning, a methodology within deep learning, holds the potential to significantly mitigate training expenses, particularly in terms of the training time required for a deep learning model. This, in turn, leads to a parallel reduction in energy consumption and the monetary investment associated with hardware components. Essentially, a model can be initially trained on a more powerful Graphics Processor Unit (GPU), subsequently shared, and eventually deployed on less powerful hardware for practical use. This technique has its roots in computer vision, where segments of a neural network — such as convolutional filters — particularly those focused on foundational image features like edges or contrast can be transposed to other models or even datasets from diverse domains. I conducted an extensive study to explore whether this effect is also transferable to IMU data, see 5.1 or [95]. While the principle appears sound in theory, its applicability to sensor-based time-discrete signals requires

further investigation due to the inherent complexity of data obtained from inertial measurement units.

Another technique frequently employed to enhance a model's classification capabilities is Data Augmentation. This technique proves particularly advantageous when a class has a limited number of instances and is consequently underrepresented. This imbalance can result in a biased model with constrained proficiency in classifying such underrepresented categories. Instances of these underrepresented classes are often found within the category of sporadic classes. In the context of basketball, notable examples include actions like *jumping* or *passing*, as discussed in Section 4.2. Employing data augmentation techniques to expand the pool of viable samples for training deep neural networks is a well-established practice within deep learning. Nonetheless, this approach warrants additional exploration, particularly concerning the determination of which dataset samples should undergo augmentation and the seamless integration of the newly generated data into the existing dataset. This is especially important when dealing with personalized IMU data. My publication [94] focuses on this topic and compares different methods with each other. This topic becomes more important when we put it in the context of long-term and real-world studies. Depending on the annotation method, samples can easily be labeled false [96] or 3.2. A model performance tends to decrease if misclassified samples are used during the augmentation phase and saved back in the dataset for training purposes which would finally lead to a poisoned classifier.

Indeed, this thesis investigates a range of research questions at the core of deep learning for wearable sensor data.

1.1.1. Principal Research Questions

RQ #1. Accurate and robust human activity recognition relies heavily on the quality of the collected dataset. From the data collection process to the annotation and analysis, multiple factors can introduce noise, biases, and errors that propagate through the modeling pipeline. We, therefore, aim to identify and mitigate such issues in activity recognition datasets. First, I examine the impact of annotation quality. Manual annotations of sensor data are required to train and evaluate activity recognition models. However, the annotation process is prone to subjectivity, inconsistency, and errors. Next, I discuss the importance of synchronized sensor streams in multi-modal activity recognition. Slight misalignments between data sources like multiple inertial sensors can significantly degrade model performance. Finally, I explore the Hawthorne effect in activity recognition - where participants alter their behavior during data collection in response to being observed and recorded. I present research quantifying this observer effect for different activities, sensor modalities, and experimental protocols. In summary, I investigate:

"How does a participant's interaction during data recording affect the outcome of a recorded dataset with regard to the quality and quantity of recorded samples and associated annotations?"

RQ #2. Recognizing complex, sporadic human activities remains an open challenge in the field of activity recognition. Most research has focused on simple, periodic activities like walking or climbing stairs. However, being able to detect activities with greater temporal and contextual variability is critical for many real-world applications. Sports analytics is one domain that would greatly benefit from improved recognition of intricate movements and plays. In this work, I explore the capabilities and limitations of current activity recognition techniques for complex activities, using basketball as a case study. Basketball provides a rich setting with frequent changes in pace and direction between players. Activities range from dribbling and shooting to running and jumping.

"What are the capabilities and limitations of current deep learning architectures for recognizing complex, sporadic, and periodic activities?"

I explore these possibilities and limitations using basketball activities as an example. Presented results provide insights into the progress and remaining challenges of complex human activity recognition, with basketball as an instructive testbed for pushing algorithms toward greater flexibility and contextual reasoning.

RQ #3. Transfer learning has shown promise for improving model performance in many domains by pre-training on large datasets before fine-tuning to a target task. In human activity recognition, applying transfer learning has achieved varying degrees of success. In this work, I aim to quantify the adaptability of transfer learning to human activity recognition. Additionally, I explore data augmentation through realistic synthetic activity data as another means to improve model generalization. Specifically, I investigate the research question:

"To what extent is transfer learning adaptable to human activity recognition data, and how can we successfully generate realistic synthetic sensor-based activity data to significantly augment a model's capabilities?"

By evaluating transfer learning strategies and generative data augmentation techniques on diverse activity recognition tasks, I provide insights into how to effectively leverage external datasets and knowledge to boost model capabilities in this domain.

In addition to the three primary research questions, the introduction of each chapter highlights supplementary secondary research questions that were addressed in the publications included in this thesis.

Section 1.2 Contributions

The contributions of this thesis can be broadly grouped into 3 topics - (1) User Studies on Human Activity Recognition, (2) Basketball Activity Recognition, and (3) Deep

Learning for Human Activity Recognition.

Chapter 3 contains publications centered on user studies on Human Activity Recognition.

Section 3.1 introduces the Activate system. An open-source data recording system with which recordings in uncontrolled and controlled environments are possible. It consists of an operating system for the Bangle.js version 1, an app, a backend, and a database.

Section 3.2 addresses the uncertainty of labeled data while conducting studies in real-world situations or settings outside the laboratory, where the deployment of video cameras to recover the ground truth is not always possible. The study compares 4 different annotation methods for data recorded in-the-wild: (1) A self-recall diary, (2) ad-hoc labeling with on-device buttons, (3) labeling with an app, and (4) a self-recall diary with visualized time-series data assistance. I found out that (4) has the best results with regard to consistency, correctness, workload for the participant, and usability.

Section 3.3 covers the importance of synchronizing multiple sensors worn by the same participants when a study is conducted with a multi-modal setup. Exemplary for other sensors, I synchronize two sensors with each other, one worn at the wrist and the eSense, a prototypical sensor worn as an earplug, using the accelerometer signals of both sensors. The algorithm achieves a precision of only 0.30 seconds of mismatch under certain constraints. Additionally, I present a study that quantifies the wearing comfort of the eSense earplug.

Section 3.4 investigates the Hawthorne Effect from the sensor data perspective. This effect indicates that people alter their behavior when being observed. Although this phenomenon is acknowledged in various human-centered studies, there is a distinct lack of comprehensive quantitative investigations regarding its influence on data quality and objectivity in monitored versus unmonitored settings, particularly in the context of Human Activity Recognition. The examination involves a combination of classical feature analysis and deep learning techniques applied to accelerometer data from ten participants. The results indicate that sensor data recorded under monitored conditions do not differ significantly from data recorded while not being monitored.

Chapter 4 presents two studies that focus on Basketball Activity Recognition. **Section 4.1** contributes a feasibility study that shows that fine-grained activity recognition is viable even in highly dynamic application scenarios like playing basketball. **Section 4.2** contributes the Hang-Time HAR dataset. A comprehensive dataset with basketball activities performed by 24 participants in two different countries.

Chapter 5 focuses on the Deep Learning aspects of this thesis by further investigating Transfer Learning (**Section 5.1**) and Data Augmentation (**Section 5.2**) for human activity data.

In **Section 5.1**, I present a study that examines the effects of Transfer Learning on a deep learning model, specifically focusing on the DeepConv-LSTM [159]. This investigation involves transfers between various sensor locations, sensor types, and even transfers between datasets from distinct domains. Results indicate that the

success of such a model transfer is very dataset and parameter-dependent and not generalizable.
Turning to **Section 5.2**, my research concentrates on diverse data augmentation strategies using the Physical Activity Monitoring for Aging People (PAMAP2) dataset [173]. I systematically examine how the process of selecting class instances for augmentation and the strategy for reintegrating them into the dataset impact the overall performance of a deep learning model.

Section 1.3

Thesis Outline

This thesis is organized in the following way:

Chapter 1: Introduction
This chapter covers the Motivation (Section 1.1), explains the Contributions (Section 1.2), and gives a brief overview of this thesis in the Thesis Outline (Section 1.3).

Chapter 2: Background and Related Work
This chapter provides comprehensive explanations of essential fundamentals within the domains of human activity recognition (**Section 2.1.1**), as well as machine or deep learning (**Section 2.2.4**). Additionally, an aggregated compilation of related work pertinent to subsequent chapters is presented in **Section 2.2**.

Chapter 3: User Studies on Human Activity Recognition
This chapter encompasses four user studies pertaining to Human Activity Recognition, with a secondary emphasis on their implications for deep learning projects.

Chapter 4: Basketball Activity Recognition
This chapter presents two studies with regard to basketball activity recognition. The first study (Section 4.1) focuses on the feasibility of applying machine learning algorithms and activity recognition methodologies to highly dynamic activities from the sports domain. The second study introduces the Hang-Time HAR dataset (Section 4.2). A publicly available, comprehensive basketball activity dataset.

Chapter 5: Deep Learning
This chapter presents studies that are focused on deep learning methodologies like Transfer Learning (Section 5.1) and Data Augmentation (Section 5.2).

Chapter 6: Conclusion and Future Work
This chapter presents the Conclusion (Section 6) and possible Future Works with regard to all presented studies (Section 6.2).

Chapter 2

Background and Related Work

This chapter will cover important background information related to all publications included in this thesis. This includes explanations regarding used hardware and sensors for recording data and an introduction to Human Activity Recognition and Deep Learning. Afterward, current scientific publications related to Human Activity Recognition and Deep Learning will be put in context for this doctoral thesis.

Section 2.1

Background

2.1.1. Human Activity Recognition

The term Human Activity Recognition (HAR) describes a discipline in Computer Science and Human-Computer Interaction where an activity performed by a human individual is recognized with regard to sensor signals that were recorded using wearable devices. This is contrasted by Action Recognition which comes historically seen from the Computer Vision. Hereby, video footage of humans carrying out activities is recorded and further analyzed. Typically, HAR researchers often use signals recorded by IMUs, which can but are not limited to include signals of accelerometers, gyroscopes, and magnetometers. Additionally to these sensors further input modalities are possible, e.g. Photoplethysmography (PPG), air pressure, or (skin) temperature sensors. An activity can be nearly everything that can be described by movement or execution patterns visible in signal recordings. This can be something sedentary, like sitting or standing, or something very complex like performing sports-specific activities that can even consist of multiple activities executed successively. Therefore, the literature defines different types of activities. Huynh categorizes human activities in 3 different categories [99]: (1) Gestures, Motions, Motifs - e.g. taking a step or bending an arm; (2) Low-level activities - e.g. sequences of movements like walking, sitting, running and (3) High-level activities/scenes/routines - e.g. sightseeing, desk work. However, other modes of categorization are common and may be more suitable, depending on

the specific domain of the study. For instance, given that this Ph.D. thesis is centered on the field of Sports Activity Recognition, the definition provided by Bock et al. [33] encompasses the entire range of activities we have investigated. The authors classify activities also in three categories, the categories are (1) Periodic activities, e.g. walking, standing, or running; (2) Sporadic activities like passing a ball or jumping and (3) Complex activities like performing a layup in basketball or a dig in volleyball. Complex, because such activities consist of multiple activities that are either performed at the same time or successively, like running, followed by jumping in an upward/forward direction and throwing the ball in the basket during a basketball *layup*. As our experiments show, such activities might be very difficult to recognize reliably.

Activity Recognition Challenges. According to Bulling *et al.* [45] activity recognition shares research challenges with other deep learning disciplines, such as Computer Vision or Natural Language Processing (NLP). For example, *Intraclass Variability*, *Interclass Similarity*, and the *NULL Class Problem*. However, HAR also has unique challenges, which are *(a) Class Imbalance, (b) Ground Truth Annotation, and (c) Data Collection Challenges*.

(a) *Class Imbalance:* Class imbalance refers to a situation where the distribution of classes within a dataset is skewed. In the context of Human Activity Recognition, certain activities may have significantly more examples than others, leading to biased models. Techniques like resampling and using specialized evaluation metrics are common strategies to address this challenge.

(b) *Ground Truth Annotation:* Ground truth annotation involves manually labeling data with the correct activity labels. In HAR, this process can be time-consuming, expensive, and subjective. Inaccurate or inconsistent annotations can impact model training and evaluation. Ensuring high-quality annotations is crucial.

(c) *Data Collection Challenges:* Collecting accurate sensor data for HAR presents challenges. Proper sensor placement, calibration, and noise reduction are essential. Capturing diverse activities, scenarios, and environmental conditions requires careful planning. Real-world variability adds complexity to dataset creation.

Furthermore, Bulling *et al.* mention so-called *Application Challenges* which refer to challenges that are connected to the hardware and recording techniques used in the experiments. Depending on the device used to record data, sensor characteristics can vary significantly and therefore often datasets recorded with different sensor modalities are difficult to transfer. In addition to that, obtaining ground truth heavily depends on the type of study. Long-term studies and studies in real-world environments have different requirements than studies conducted in a controlled environment. Addressing these challenges involves careful experimental design, preprocessing, annotation strategies, and the application of suitable machine-learning techniques.

Recording Environments. The recording environments influence the characteristics of activities. For example, running in real-world conditions will differ from running on the treadmill. Therefore, we distinguish between controlled or lab environments and uncontrolled or real-world/in-the-wild environments. Both recording environments come with their advantages/disadvantages and research challenges. Often, ground truth is obtained in hindsight by filming the participants during their exercises if a dataset was recorded in the lab. However, under uncontrolled conditions obtaining ground truth can be a challenge by itself, since the installation of devices like video cameras is often simply not possible. Furthermore, studies have shown that participants tend to alter their behavior as soon as they notice that they are recorded. Therefore, such circumstances can further influence data characteristics. These differences can be reflected in machine learning models that are trained with data from one specific environment and therefore assumed that a classifier trained on data recorded in-the-wild won't be able to classify data from a lab-made dataset with high confidence.

Activities: Periodic Activities. Running, walking, standing, and laying are the classical periodic activities that can be found in many datasets with respect to Activities of Daily Living (ADL). Such activities have in common that the signals oscillate in constant patterns as long as the participant does not alter his/her motorics significantly. These activities can be learned and recognized comparatively easily by Deep Learning models due to their low complexity, for which simple convolutional layers are usually sufficient.

Activities: Sporadic Activities. Sporadic activities are activities that are performed occasionally and do not necessarily follow a repetitive pattern. Taken from the field of basketball activity recognition, such activities can be passing or rebounding a ball. The moment of execution of sporadic activities is highly context-dependent and does therefore not occur regularly if the data were recorded in uncontrolled environments. Therefore, instances of these classes in datasets are normally very limited, if not a specific exercise was executed to artificially increase the number of repetitions. Hence, a simple network architecture has limited capabilities to recognize these activities.

Activities: Complex Activities. Complex activities are a category of activities that are challenging to recognize - even for state-of-the-art architectures. Such activities have in common that they consist of multiple activities that are either executed consecutively or in parallel. This increases the intra-class variability, which consequently impedes the generalizability of a model. Examples of such classes can be many household activities, like cleaning, or from the explicit field of basketball activities a shot or layup.

2.1.2. Hardware

The experiments detailed in this doctoral thesis primarily employed data collected by us through the use of three distinct devices. These devices encompass the Platypus, a self-developed wrist-worn prototype equipped with a full IMU capable of sampling at rates of up to 300 Hertz (Hz) sampling rate with a sensitivity range of ±16g. Additionally, the eSense, an ear-worn IMU developed by Nokia-Bell, integrates an accelerometer and gyroscope and can capture data at a maximum rate of 50 Hz. Furthermore, the Bangle.js, an open-source smartwatch, features an accelerometer and magnetometer, with a maximum sampling rate of 100 Hz and a sensitivity of ±8g. Zhou *et al.* [242] categorizes IMUs into 4 performance classes: (1) marine and navigation grade IMUs, (2) tactical-grade IMUs, (3) industrial-grade IMUs, and (4) hobbyist grade. (1) is mostly used for ship, air- and spacecraft navigation, the second performance class is used for unmanned aerial navigation, (3) is installed on robotics and industrial machinery and the fourth performance class is widely used in automotive or consumer grade devices like activity trackers or gaming devices. Further specifications for the devices used during our research will be described in the forthcoming chapters. The subsequent Table 2.1 presents a summary of the hardware specifications for the tools utilized during the course of this thesis.

Table 2.1 This table provides a quick look at the hardware and their specifications used in the projects of this thesis.

	Chipset	RAM	Storage	Sensorics	Max. Sensitivity (Accelerometer)	Max. Sampling Rate
Platypus (wrist-worn)	22nm Intel Atom "Tangier" (Z34XX) with 2 cores with 500 MHz, Intel Quark with single core 100 MHz	1GB	4GB	Accelerometer Gyroscope Magnetometer Ambient Light sensor	±40g	300 Hz
eSense (ear-worn)	Qualcomm CSR8670 with 80 Mhz	56 kB	None	Accelerometer	±20g	50 Hz
Bangle.js Version 1 (wrist-worn)	Nordic 64MHz nRF52832 ARM Cortex-M4	4 MB	4 MB	Accelerometer Magnetometer PPG (Skin) Temperature	±8g	100 Hz

Devices: Platypus. The bulk of the computing power, power management, and wireless communication modules is provided by this off-the-shelf board, which is produced by Intel Corporation. The EDI2.SPON.AL.S version of the module, which is used for the Platypus, is CE and FCC-certified and specifically made for wearable devices. The module's main processor is a 22nm Intel Atom "Tangier" (Z34XX) that includes two Atom Silvermont cores running at 500 MHz and one Intel Quark core at 100 MHz (for executing RTOS ViperOS). The system has 1 GB RAM integrated into the package. There is also 4 GB eMMC flash storage on board, with Wireless Local Area Network (WiFI), Bluetooth 4, and USB controllers. Its dimensions are 35.5 x 25 x 3.9 mm. The Edison module runs an embedded version of the Linux operating system, Yocto, which is an open-source collaboration project that provides templates, tools, and methods to help create custom Linux-based systems for embedded products, regardless of the specific hardware architecture. All sensors are populated

Figure 2.1 Our wrist-worn sensor prototype captures, pre-processes, and classifies data locally. It has an energy-efficient display, a full IMU, 5 sensors, a 500 MHz dual-core processor, and a 100 Hz microcontroller. It runs embedded Linux and connects via Bluetooth and WiFi.

on an Edison-compatible printed circuit board that contains several sensors that immediately interface to the Edison's microprocessors. Additionally, a battery gauge and recharging circuit are added, as well as a miniature display connector for a Sharp Memory LCD. This collection of peripheral modules is directly interfaced to the Edison board via its miniature 70-pin connector. The board has furthermore been extended to contain optical pulse oximeters or sensors for measuring skin conductivity, as separate modules attached to the custom sensor Printed Circuit Board (PCB). The prototype is powered by an off-the-shelf 3.7V, 600 mAh Lithium-Ion rechargeable battery of similar dimensions. The display is a 1.28 inch (32.51mm) 128 x 128 pixel Monochrome HR-TFT Transflective LCD Panel produced by Sharp, which is especially energy-saving when infrequently updated. The most important sensor for this paper is the MPU-9250 (by Invensense), which includes a 3D accelerometer, 3D gyroscope, and a 3D magnetometer to capture motion and orientation as accurately as possible. Figure 2 shows the whole prototype, enclosed in a custom-built case with a transparent top part so that the light sensors can still capture ambient light conditions and the display remains visible, without requiring holes.

Devices: eSense. The eSense is an ear-worn multi-sensor platform that comes with a 6-axis IMU (accelerometer and gyroscope), a microphone, dual-mode Bluetooth, Bluetooth Classic and Bluetooth Low Energy (BLE), and high-definition wireless audio speakers.

Figure 2.2 eSense prototype, developed by Nokia Bell Labs. Figure adapted from [105]

The device does not have on-device flash memory or a built-in real-time clock. However, the data can be streamed, e.g. to a smartphone, via BLE reliable with up to 50Hz. The PCB is equipped with a Qualcomm CSR8670, a programmable Bluetooth dual-mode flash audio System-on-Chip (System-on-Chip (SoC)) with one microphone; a TDK MPU6050

six-axis inertial measurement unit including a three-axis accelerometer, a three-axis gyroscope, a digital motion processor, and a two-state button; a circular LED; associated power regulation; and battery-charging circuitry, see Kawsar *et al.* [105].

Devices: Bangle.js. The Bangle.js Version 1[1] is a commercial open-source smartwatch on which our open-source firmware[2] was installed. The device comes with a Nordic 64MHz nRF52832 Advanced Reduced Instruction Set Computer (RISC) Machines (ARM) Cortex-M4 processor with Bluetooth LE, 64kB RAM, 512kB on-chip flash, 4MB external flash, a heart rate monitor, a 3D accelerometer, and a 3D magnetometer. Our firmware only uses the 3D accelerometer and provides the user with the basic functions of a smartwatch, like displaying the time and counting steps. The data were recorded with 25 Hz and ±8g and are saved on the devices' memory with a delta compression algorithm. Therefore, it is possible to save up to 8-9 hours (depending on how much of the data could be compressed) of data with the given parameters. The smartwatch stops recording as soon as the memory is full. At the end of the day, the participants need to upload their daily data and program the starting time for the next day using our upload web-tool[3].

Figure 2.3 Bangle.js 1 Smartwatch with our Activate firmware installed. The display indicates the battery charging level, the number of steps taken during the day, and the number of active minutes [104].

Sensorics: Accelerometer. The accelerometer senses the acceleration of gravity in g. In such a sensor, a medium that can swing along one specific axis is positioned between two or four springs on free bearings. If the sensor is now accelerated, this medium pushes itself in a specific direction (positive or negative direction along the axis) and thus deviates from the original position. This deviation now generates a change in the capacitance value which can be measured and interpreted accordingly.

Sensorics: Gyroscope. A gyroscope or angular rate sensor measures the rate of turn (rad/sec) without a fixed point of reference. In modern Micro Electrical Mechanical System (MEMS) a vibrating mass is installed on the device which oscillates fixed along a given axis as long as no force is applied to the sensor. However, the oscillation of the mass deviates from the fixed axis as soon as force is applied to the sensor in a certain direction due to rotation. These deviations from the norm can then be measured and used for our experiments.

Sensorics: Magnetometer. The magnetometer is another sensor often installed on IMUs. Even though this type of sensor was not used in any of the studies presented

[1] `https://www.espruino.com/banglejs`
[2] `https://github.com/kristofvl/BangleApps/tree/master/apps/activate`
[3] `https://www.ubi29.informatik.uni-siegen.de/upload`

in this manuscript, I decided to add it to the basics for the sake of completeness. The magnetometer measures the magnetic field of the earth around a body, e.g. a smartphone or a wearable device, with respect to its orientation. To do so, the so-called Hall Effect is used. When a conductive medium. For example a copper plate is put under current, the electrons will flow straight through the medium. The Hall Effect describes that if we bring a magnetic field close to that medium, we disturb the straight flow - due to the appliance of an external force, called Lorentz Force - of the electrons and further bind them to one side of the medium. On the other side of the medium, the positive charge is accumulated. These separated accumulations of negative and positive charges now generate a voltage that can be measured. Due to the variation of the earth's magnetic field depending on the orientation and location of the magnetometer, the distribution of the negative and positive charges varies. These variations can be measured and interpreted and later used to determine the orientation of a wearable device.

Deep Learning on Sensor Data. Deep learning, a subfield of machine learning inspired by the neural networks of the human brain, has exhibited remarkable proficiency in discerning intricate patterns and extracting knowledge from complex data. It effectively embodies an approximated function tasked with mapping input data to output variables. This function is acquired through the iterative process of backpropagation, wherein the network adjusts its internal parameters to minimize the discrepancy between its predictions and the actual target values in the training data.

The quality and accuracy of the learned approximation hinge upon several critical factors, including the characteristics of the training data and the configuration of various parameters. These parameters encompass learning rate, activation functions, loss functions, weight initialization schemes, regularization techniques, and optimization algorithms. Additionally, the presence of noise and uncertainty within the training data can significantly influence the model's performance.

Furthermore, optional post-processing steps may be applied to the model's output, tailoring it to meet specific requirements or constraints imposed by the application domain. Once a deep learning architecture has undergone the training process and attained a satisfactory level of performance, it is commonly referred to as a "model".

Deep learning applied to sensor data holds several distinctive challenges, requiring specialized approaches and considerations. These challenges are:

(a) *Temporal Dependencies*: Sensor data often comprises time-series information, where observations are recorded over time. Capturing and modeling temporal dependencies in the data is crucial. Techniques such as Recurrent Neural Networks (RNNs), or Temporal Convolutional Networks (TCNs) are frequently employed to address this challenge.

(b) *Multimodal Data Fusion*: Many sensor data applications involve multiple sensors of different types or modalities. Integrating information from diverse sources, such as accelerometers, gyroscopes, cameras, or environmental sensors, is essential for holistic analysis. Methods like sensor fusion, feature fusion, or multi-stream

neural networks are employed to fuse data from various sensors effectively.

(c) *Data Variability*: Sensor data can exhibit variability arising from multiple sources. Individual differences among users wearing sensors, environmental changes, or external factors can introduce variations in the data. A robust deep-learning model needs to be capable of handling and adapting to such variability.

The subdomains of *Transfer Learning* and *Data Augmentation* hold particular significance within the scope of this thesis since this research includes a series of experiments focused on these specific areas of study. We, therefore, would like to take the time to explain these two topics.

Transfer Learning is a technique initially employed in the field of Computer Vision, which transfers the knowledge gained from a model pre-trained on one dataset to another dataset within the same domain. In HAR, this typically involves pretraining a deep learning model on a source dataset, which could contain activities like walking, running, or cycling. Once the model has learned meaningful features from this source data, it is fine-tuned or adapted to a target dataset that might or might not contain different activities or variations of activities. This fine-tuning process allows the model to adapt its learned features to the specific characteristics of the target dataset. This technique offers the advantage of significantly reducing the training time required for the second model. It also aids in the adaptation of a model across different domains and proves beneficial when dealing with target datasets of limited size or lower annotation quality. Transfer Learning can encompass various aspects, including specific preprocessing or postprocessing steps applied to the data. Additionally, techniques like weight transfer are commonly employed, involving the transfer of pre-trained weights from intermediate layers, such as convolutional layers, from a source model to a target model.

Data Augmentation for Human Activity Recognition is a technique used to increase the diversity and quantity of training data by creating new, slightly modified samples from the existing dataset. It involves strategies like rotation and orientation adjustments, time warping, amplitude scaling, noise injection, temporal jittering, and more. These modifications introduce diversity into the dataset, enabling the model to better generalize and recognize activities accurately. Data Augmentation is particularly valuable when dealing with limited or imbalanced data. However, it's crucial to strike a balance to avoid introducing excessive noise and overfitting. The choice of augmentation methods depends on the dataset's characteristics and modeling goals.

Section 2.2

Related Work

The Related Work chapter is subdivided into various subtopics relevant to Human Activity Recognition. These subtopics encompass Human Activity Recognition, annotation methods, and the potential biases associated with their use, the Hawthorne Effect in HAR, and Human Activity Recognition in Sports Science. Human Activity Recognition in Sports Science, explores the application of wearable technology in the context of sports-related activities. Furthermore, this chapter extensively discusses publications pertinent to Deep Learning and Machine Learning, with a thematic focus on Transfer Learning and Data Augmentation. By organizing the chapter in this manner, the dissertation provides a comprehensive and well-structured review of the existing literature, laying the foundation for the subsequent chapters' exploration and analysis.

2.2.1. Activity Recognition

One of the utmost responsibilities in ubiquitous or pervasive computing is delivering precise and timely information concerning individuals' activities and behaviors. The potential applications are limitless, spanning from healthcare and geriatric care to sports monitoring and industrial uses, encompassing various sectors like process optimization, robotics, automotive, and defense technology. Sensor technology and activity recognition have become integral parts of people's daily lives, especially with the widespread adoption of various wearable devices.

Application Environments. The application environments for analyzing sensor data are diverse and encompass various aspects of daily life for many individuals. This subsection aims to categorize several significant ones, ranging from monitoring daily activity routines to working environments, such as industrial settings. Additionally, sports monitoring applications and medical applications, like elderly care, assisted living, and supporting medical operations, fall within this scope.

The recognition of patterns based on sensor data finds valuable applications in **industrial** settings. Examples include predictive maintenance [171] and assisting workers during specific manufacturing processes, like welding or bending workpieces [141]. In such environments, the prevalent use of full IMUs for activity recognition is less common. Instead, the focus shifts towards employing environmental sensors, such as temperature and air pressure sensors, as well as vibration sensors based on accelerometers. An intriguing and less conventional example in the realm of industrial applications is the Skoda mini checkpoint dataset, as presented by Zappi *et al.* in 2008 [237]. This dataset captures the activities of an assembly line worker within a car manufacturing plant in Skoda.

Sports represent a major application domain for Human Activity Recognition due to the vast diversity of athletic activities and their importance for fitness tracking,

training optimization, injury prevention, and performance analysis. HAR techniques have been applied to recognize activities in various sports, such as swimming [44], **basketball - including our publications [89, 93] -** and soccer [196]. A detailed overview can be found in Table 2.2. By detecting granular motion patterns, HAR enables detailed quantification of biomechanics, physiology, and performance indicators that can provide athletes and coaches with unprecedented insights [106]. For instance, inertial sensor-based HAR can identify improper exercise forms that may lead to injury (Acharya et al., 2018). HAR can also automatically track training load, exercise frequency, and other variables to optimize training regimens and competition readiness (Falcone et al., 2020). Furthermore, integrating HAR with contextual data like heart rate provides a richer assessment of physiological demands during training and competition (Wang et al., 2018). In summary, the diversity of athletic activities coupled with the utility of motion analytics makes sports a significant domain for applying HAR techniques. This thesis focuses mainly on sports activity recognition, therefore, a more detailed section on related work on sports activity recognition

The **healthcare sector** is an important field for applying Human Activity Recognition systems for a variety of patient monitoring and care enhancement applications. HAR uses wearable sensors and machine learning algorithms to automatically detect human activities, which can provide ubiquitous, unobtrusive monitoring in both clinical and home settings. For example, HAR systems can be used to continuously monitor elderly or chronic disease patients to detect falls, track medication adherence patterns, monitor rehabilitation progress, and identify emerging health issues without constant supervision [5]. In assisted living environments, HAR enables smart home systems to support independent living by detecting activities of daily living and alerting caregivers if anything seems amiss, such as a lack of movement in the morning indicating a potential overnight fall. HAR can also analyze staff workflows in hospitals to identify inefficient patterns and improve healthcare operations. The data quantified by HAR systems can be used to assess patient recovery, such as analyzing gait characteristics to monitor hip surgery rehabilitation progress, enabling clinicians to better tailor therapies. Consumer wellness applications like fitness trackers and sleep monitoring apps also rely on HAR algorithms to automatically detect exercises, sleep quality, and daily behaviors so personalized health insights can be provided to users. In summary, HAR provides the ubiquitous sensing capabilities to continuously monitor patients, assist the elderly and impaired, quantify recovery, and detect wellness indicators that can enhance the quality of care and improve outcomes by enabling personalized, data-driven healthcare.

Daily routines is an interesting and useful application area for Human Activity Recognition systems because the area encompasses many common recurring activities that people perform regularly, such as sleeping, eating, chores, exercise, work, etc. By being able to automatically detect and analyze patterns in these routine activities using sensors and machine learning algorithms, HAR enables a wide range of assistive technologies and wellness applications. For example, deviations from a person's normal daily activity patterns could provide indicators of emerging health issues or

behavior changes. HAR can also quantify metrics like step counts, activity levels, and sleep duration to allow people to self-monitor their fitness goals and maintain general wellness. In smart home environments, recognizing activities allows automated systems to control lighting, heating, ventilation, air conditioning, and appliances in ways that suit the user's habits and preferences. If a HAR system detects that a person is cooking, it can turn on the ventilation fan or play their favorite music. The contextual awareness provided by recognizing ongoing activities also enables systems to deliver tailored reminders and assistance, like notifying someone to take their medication when they wake up in the morning.

Annotation Methods in Activity Recognition. According to Stikic *et al.* [194] and later Cleland *et al.* [56], we distinguish between 6 or 7, respectively, different methods and 2 environments (online/offline) of labeling data, the methods are (1) Indirect Observation, (2) Self-Recall, (3) Experience Sampling, (4) Video/Audio Recordings, (5) Time Diary, (6) Human Observer, (7) Prompted Labeling. Cruz *et al.* [57] uses 4 different categories to classify data labeling approaches, these are **(1) temporal (when)** - is the label conducted during or after the activity, **(2) annotator (who)** - is the label given by the individual itself or by an observer, **(3) scenario (where)** - is the activity labeled in a controlled (e.g laboratory) or uncontrolled (in-the-wild) environment, and **(4) annotation mechanism (how)** - is the activity labeled manually, semi-automatically or fully-automatically. All labeling methods have their benefits and costs and come with a trade-off between required time and label accuracy. However, not every method is suitable for long-term and in-the-wild recording data. Reining *et al.* [172], evaluated the annotation performance between 6 different human annotators of a Motion Capturing (MoCap) (Motion Capturing) and IMU HAR Dataset for industrial deployment. They concluded that annotations were moderately consistent when subjects labeled the data for the first time. However, annotation quality improved after a revision by a domain expert. In the following, I would like to go into more detail on what I consider to be the most important labeling methods for the specific field of activity recognition.

2.2.2. Annotation Methods in Activity Recognition

Self-Recall methodologies are generally called methods in which study participants have to remember an event in the past. This methodology is used, for instance, in the medical field (e.g. in the diagnosis of injuries [214]), but also frequently in studies in the field of long-term activity recognition. Van Laerhoven *et al.* [217] used this method during a study in which participants were asked to label their personal daily data at the end of the day. They noticed that the label quality depends heavily on the participant's recall and can therefore be very coarse. During a study conducted by Tapia *et al.* [200], every 15 minutes a questionnaire was triggered in which participants needed the answer multiple choice questions about which of 35 predefined activities were recently performed.

App Assisted Labeling: Cleland *et al.* [56] presented in 2014 the so called Prompted

labeling. An approach that is already used by commercial smartwatches like the Apple Watch[4]. In this study, users were asked to set a label for a time period that has been detected as an activity right after the activity stopped. Akbari *et al.* [9] leverages freely available Bluetooth Low Energy (BLE) information broadcasted by other nearby devices and combines this with wearable sensor data in order to detect context and direction changes. The participant is asked to set a new label whenever a change in the signal is detected. Gjoreski *et al.* [73] published 2017 the SHL dataset which contains versatile labeled multimodal sensor data that has been labeled using an Android application that asked the user to set a label whenever they detected a position change via GPS. Tonkin *et al.* [207] presented a smartphone app that was used in their experimental smart home environment with which study participants were able to either use voice-based labeling, select a label from a list of activities ordered by the corresponding location or scan Near Field Communication (NFC) tags that were installed at locations in the smart house. Similar to Tonkin *et al.* [207], Vaizmann *et al.* [213] developed an open-source mobile app for recording sensor measurements in combination with a self-reported behavioral context (e.g. driving, eating, in class, showering). 60 subjects participated in their study. The study found that most of the participants preferred to fill out their past behavior through a daily journal. Only some people prefer to set a label for an activity that they are about to do. Schröder *et al.* [182] developed a web-based GUI that can be used on a smartphone, tablet, or PC to label data recorded in a smart home environment. However, it is important to mention that, According to Cleland *et al.* [56], the process of continually labeling data becomes laborious for participants and can result in a feeling of discomfort.

Unsupervised Labeling is a methodology that uses clustering algorithms to first categorize new samples without deciding yet to which class a sample belongs. Leonardis *et al.* [117] presented in 2002 the concept of finding multiple subsets of eigenspaces where, according to Huynh [99], each of them corresponds to an individual activity. Huynh uses this knowledge to develop the eigenspace growing algorithm, whereby, *growing* refers to an increasing set of samples as well as to increase the so-called *effective dimension* of a corresponding eigenspace. Based on the reconstruction error (when a new sample is projected to an eigenspace), the algorithm tries to find the best-fitting representation of a sample with minimal redundancy. Hassan *et al.* [86] recently published a methodology that uses the Pearson Correlation Coefficient to map very specific labels of a variety of datasets to 4 meta labels (inactive, active, walking, and driving) of the ExtraSensory Dataset [213].

Human-in-the-Loop (Labeling) is a collective term for methodologies that integrate human knowledge into their learning or labeling process. Besides, being applied in HAR research, such techniques are often used in Natural Language Processing and according to [227] the NLP community distinguishes between entity extraction [71, 240], entity linking [108], Q&A tasks [220] and reading comprehension tasks [23].

Active Learning is a machine learning strategy that currently receives a lot of attention in the HAR community. Such strategies involve a Human-in-the-Loop for

[4] `https://www.apple.com/watch/`

labeling purposes. In the first step, the learning algorithm automatically identifies relevant samples of a dataset which are posteriorly queued to be annotated by an expert. Incorporating a human guarantees high-quality labels which directly leads to a better-performing classifier. Whether a sample is determined to be relevant, as well as the decision to whom it may get presented for annotation purposes is the main focus of research in this field. Bota *et al.* [38] presents a technique that relies on specific criteria defined by 3 different uncertainty-based selection functions to select samples that will be presented to an expert for labeling and then propagated throughout the most similar samples. Adaimi *et al.* [6] benchmarks the performance of different Active Learning strategies and compares them, with regard to 4 different datasets with a fully-supervised approach. The authors concluded that Active Learning needs only 8% to 12% of the data to reach similar or even better results than a fully-supervised trained model. These results suggest that presenting pre-selected samples to a human for labeling purposes can reduce the amount of data needed to train a machine learning classifier significantly due to the increased quality of the labels. Miu *et al.* [143] presented a system that used the Online Active Learning approach published by Scully [185] to bootstrap [4] a machine learning classifier. The publication presented a smartphone app that asked the user right after finishing an activity, which activity has been performed. Afterward, a small subset of the labeled data was used to bootstrap a personalized machine-learning classifier.

The Hawthorne Effect. With researchers from a multitude of human-centered studies being aware of the existence of such an effect, a data-driven study of the phenomenon and its potential effects remain largely under-explored in the community of Human Activity Recognition. Human Activity Recognition typically relies on participants being monitored via wearable sensors, making them consistently aware of being observed. However, these circumstances may have introduced a behavior bias [234] into publicly available datasets. This bias manifests as changes in behavior when study participants are aware of being monitored by another person or a video recording system [69]. The fundamental research of which the Hawthorne effect originated, was conducted between 1924 and 1927 as part of an investigation of whether the productivity of workers of the Hawthorne Western Electric plant could be increased by a change in lighting conditions [138]. With later studies criticizing the research methodology [49], Landsberger concluded in 1958 [112] that the increase in productivity was to be attributed to the workers being aware that they were monitored and not the change in working conditions. The observed phenomenon, i.e. the alteration of behavior whenever participants are aware that they are being monitored, was later then primed as the Hawthorne Effect. In the context of HAR experiments, the Hawthorne Effect suggests that participants' awareness of being monitored can potentially affect the applicability and generalization of trained activity recognition systems. Knowing they are being observed and activities are being recorded, participants might result in modifying their movements, behaviors, and/ or daily routines, leading to a deviation from a natural execution of activities. This alteration

can introduce biases and inaccuracies in the data collected for HAR experiments, making it challenging to develop reliable and scalable activity recognition systems [49]. To mitigate the Hawthorne Effect in HAR experiments, researchers often opt for minimizing participants' awareness of being monitored. By employing discrete sensing techniques, such as using a minimalistic setup of wearable devices [213] or ambient sensors [144], they can collect activity data without participants constantly focusing on the monitoring process. By reducing the conscious attention given to monitoring, researchers aim to capture more natural and representative data that can improve the accuracy and reliability of HAR systems [96]. While the Hawthorne effect has been demonstrated psychologically, no scientific publication has examined it from a sensor perspective. Our publication [90] aims to highlight this aspect and further investigate how a participant's awareness of being observed may influence data quality and classifier performance.

In 2014, Bulling *et al.* [45] described several research challenges in creating datasets for Human Activity Recognition that avoid bias. These challenges, being still relevant to this date, include *intra-class* variability, *inter-class* similarity, and the *NULL-class* problem. Important in the context of the Hawthorne Effect is the *intra-class* variability, which describes how data from the same class differ between participants or sometimes even instances of one activity from the same individual due to stress, fatigue, or an emotional or environmental state in which the activity is performed. As such, the Hawthorne Effect can be categorized as an *intra-class* variability problem - which can have a direct effect on classifier capabilities and performance.

2.2.3. Sports Activity Recognition

IMU-based sports activity recognition is one of the main application fields for HAR studies, as summarized in Table 2.2 and Table 2.3. It has already been proven for a variety of different sports, such as running [24, 149, 178], ball sports [235, 196, 72, 47, 131, 37], winter sports [115], sports for the disabled [85] or fitness [160, 201, 59], that activity recognition algorithms are capable of detecting specific activities tied to these sports based on IMU data as input. Basketball has been used by several studies based on IMU sensor data since 2014. The studies presented in Table 2.3 focused on a wide field of applications within basketball.

Hoelzemann *et al.* [89] detected different dribbling styles and shooting the ball with one single wrist-worn full IMU and later on published the Hang-Time HAR dataset [93], both publications are presented in detail in Chapter 4. Mangiarotti *et al.* [135] used two IMUs worn on both wrists to differentiate between passing, shooting, and dribbling the ball. Sviler *et al.* [197] focused on locomotion-bound activities, like jumping, acceleration, deceleration and change of direction. Sangüesa *et al.* used IMU and Red, Green, Blue (RGB) video data to detect complex basketball tactics. Lu *et al.* [126] and Liu *et al.* [123, 124] attached smartphones to the body and used the built-in accelerometer to detect a variety of basketball activities. Lu *et al.* and Liu *et al.* showed that accelerometer data alone is sufficient to classify basketball activities. Technically wise, the most comprehensive

Table 2.2 IMU-based studies have been performed throughout many different sports in the past years, yet few are publicly available for usage by other researchers (table partially based on [15], Table 2).

Sports Studies with Wearables

Study	Sport & (#) Activities Performed	Sensors/Systems Used	# Subjects	Published	Analysis Method
Bastiaansen *et al.* [24]	(1) Sprinting	Five IMUs and sensor fusion algorithms	5	No	Statistical Analysis
Borja Muniz-Pardos *et al.* [149]	(1) Running	Foot worn inertial sensors	8	No	Statistical Analysis
Brouwer *et al.* [42]	(5) Swing motions from different sports: golf swings, 1-handed ball throws, tennis serve, baseball swings. and a variety of trunk motions.	Two IMUs and a MoCap system	10	No	Statistical Analysis
Brunner *et al.* [44]	(5) Swimming	Wrist-worn full IMU, barometer	40	No	Deep Learning (CNN)
Carey *et al.* [47]	(1) Physical impacts while playing rugby	head-worn accelerometer and gyroscope (x-patch™)	8	No	Statistical Analysis
Lee *et al.* [115]	(2) Skiing turns	17 IMUs and pressure sensors	7	No	3D Kinematic Model Evaluation
Teufl *et al.* [201]	(3) Bilateral squats, single leg squats, and counter-movement jumps	Seven IMUs and a MoCap system	28	No	Rigid Marker Cluster, Statistical Analysis
Wang *et al.* [222]	(3) Racket Sports	Wrist-Worn IMU	12	No	Machine Learning, (SVM, Naive Bayes)
Whiteside *et al.* [224]	(9) Tennis strokes	Wrist-Worn IMU	19	No	Statistical Analysi
Ghasemzadeh and Jafari [72]	(1) Baseball swing	3 IMUs (Wrist, Shoulder, Hip)	3	No	Semi Supervised Clustering
MacDonald *et al.* [131]	(15) Volleyball	6D IMU (Acc. & Gyr.)	13	No	Statistical Analysis
Borges *et al.* [37]	(6) Volleyball	Waist worn full IMU	112	No	Statistical Analysis
Dahl *et al.* [59]	(5) Cutting, running, jumping, single leg squats and cross-over twist	8 full IMUs, 17 MoCap Cameras	49	No	Statistical Analysis
Pajak *et al.* [160]	(4) Fitness exercises: dips, pullups, squats, void	3 full IMUs, Pressure Sensor, Radio Signal	-	No	Deep Learning (CNN)
Yu *et al.* [235]	(1) Soccer kick	6D IMU (Acc. & Gyr.)	-	Yes, upon request	Attitude Estimation with Quaternions, Gravity Compensation
Stoeve *et al.* [196]	(3) Soccer kick, pass, void	Shoe-worn IMU	836	No	Machine and Deep Learning (SVM, CNN, DeepConv-LSTM)
Bock *et al.* [36]	(19) Fitness activities	4 Accelerometer sensors, egocentric video footage	18	Yes	Deep Learning (DeepConv-LSTM, Attend-and-Discriminate, ActionFormer)
Brognara *et al.* [41]	(-) CrossFit®	Full IMU at the lower back	42	Yes, upon request	Statistical Analysis
Perri *et al.* [165]	(8) Tennis strokes	1 Full IMU at the scapulae	8	Yes, upon request	Statistical Analysis
Azadi *et al.* [17]	(1) Alpine skiing	2 smartphones with IMUs placed at the pelvis	11	No	Unsupervised Machine Learning (Gausian Mixture Models, Kmeans)
Jean *et al.* [103]	(-) Running	foot-worn 6-axis IMU	41	No	Statistical Analysis
Yang *et al.* [230]	(-) Contact and flight-time (Running)	2 ankle-worn 6-axis IMUs	36	Yes, upon request	Statistical and Feature Analysis
Léger *et al.* [116]	(3) Ice Hockey	1 glove-worn IMU	10	Yes, upon request	Machine Learning (kNN)
Hamidi *et al.* [81]	(-) Swimming performance	1 sacrum-worn IMU	15	Yes, upon request	Statistical Analysis, Self-Assessment
Müller *et al.* [147]	(-) Beach Handball performance	1 full IMU placed at the upper thoracic spine	69	Yes	Statistical Analysis
Patoz *et al.* [164]	(-) Contact and flight-time (Running)	1 sacral-mounted IMU	100	Yes, upon request	Statistical Analysis
Lee *et al.* [114]	(4) stride, step, and stance duration of running gait	Sacrum worn 3D Accelerometer, 6 infrared cameras	10	No	Statistical Analysis
Harding *et al.* [84]	(-) Airtime analysis of snowboarders	One 3D gyroscope	10	No	Statistical Analysis

basketball activity study so far was conducted by Nguyen *et al.* [155]. The group used data from 5 full IMUs attached to the participants' shoes, knees, and lower back to classify frequently occurring basketball activities like walking, running, jogging,

pivot, jumpshot, layupshot, sprinting, and jumping. However, Table 2.3 shows that the only study that made their dataset publicly available is Trost *et al.* [208], even though this dataset is not available for download at the moment when this manuscript is written. Observing a basketball game and interpreting activities executed on the court is a research topic majorly driven forward by computer vision studies. Therefore,

Table 2.3 IMU-based basketball activity recognition studies. Trost *et al.* is the only team that made their dataset publicly available for download. However, the dataset is currently unavailable for download. (*Not accessible from the source given by the manuscript at the time of writing.)

Sensor Based Basketball Studies

Study	(#) Activities Performed	Sensors/Systems Used	# Subjects	Published	Analysis Method
Hoelzemann *et al.* [89]	(4) different dribbling techniques, shooting	Wrist-Worn Full IMU	3	No	Machine Learning (kNN, Random Forest)
Sviler *et al.* [197]	(4) jumping, acceleration, deceleration and change of direction	Full IMU	13	No	Statistical Analysis
Nguyen *et al.* [155]	(8) walking, running, jogging, pivot, jumpshot, layupshot, sprinting, jumping	Five Full IMUs	3	No	Machine Learning (SVM)
Trost *et al.* [208]	(7) lying, sitting, standing, walking, running, basketball, dancing	Two Full IMUs,	52	Yes*	Statistical Model (Logistic Regression Model)
Bo [32]	(5) standing, running standing dribble, penalty shot, jump shot	5 IMUs (Acc. & Gyr.)	20	No	Deep Learning (RNN)
Lu *et al.* [126]	(5) standing, bouncing ball, passing ball, free throw, moving with ball	3 smartphones with accelerometer	4	No	Multiple Supervised Machine Learning Classifier
Liu *et al.* 2015 [123] and 2016 [124]	(8) walk, run, jump, stand throw ball, pass ball, bounce ball, raise hands	2 smartphones with accelerometer	10	No	Multiple Supervised Machine Learning Classifier
Sangüesa *et al.* [179]	(5) complex basketball tactics: (pick and roll, floppy offense press break, post up, fast break)	IMUs and video footage	11	No	Machine Learning (SVM)
Mangiarotti *et al.* [135]	(3) passing, shooting, dribbling	two wrist-worn IMUs	2	No	Machine Learning (SVM, kNN)
Staunton *et al.* [193]	(1) jumping	MARG Sensor (magnetic, angular rate and gravity).	54	No	Statistical Analysis
Eggert *et al.* [65]	(1) jump shot	foot-worn IMU	10	No	Deep Learning (CNN)
Bai *et al.* [19]	(1) basketball shots	one wristband-worn IMU, one Android smartphone put in the trouser pocket.	2	No	Multiple Supervised Machine Learning Classifier
Hasagawa *et al.* [85]	(2) Wheelchair basketball: push and stop	wheelchair equipped with two IMUs	6	No	Feature and Statistical Analysis

according to Table 2.2.3 Computer Vision-based activity or action recognition datasets are already publicly available widely to the community. The datasets presented in Table 2.2.3 mostly contain RGB data. The dataset used by Hauri *et al.* [87] is available for download and contains, among other modalities 1D (y-axis) accelerometer data of National Basketball Association (NBA) players shooting a basketball. However, the authors confirmed to us that the acceleration data in their dataset were not recorded with a wearable sensor device. Moreover, they were extrapolated from the video data by taking into account the positional data of the players and the time stamps. The study focused on detecting complex tactical group activities like pick and roll or handoffs. The studies conducted with visual data are more comprehensive with regards to the number of classes that are distinguished between, compared with IMU-based activity studies. Gu *et al.* [80] classified 26 fine-grained basketball activities into 3 broad categories. A very early study, conducted in 2008 by De Vleeschouwer *et al.*

Table 2.4 Vision-based basketball activity recognition studies that published their dataset for download. However, Maksai *et al.* and Ramanthan *et al.* are currently not available for download. (*Not accessible from the source given by the manuscript at the time of writing.)

Vision-based Basketball Studies

Study	Action Recognized	Sensors/Systems Used	Published
Hauri *et al.* [87]	Group activities: pick and roll, handoff	Videos and 1D-Accelerometer (only shots, extrapolated from videos)	Yes
Ma *et al.* [128]	12 atomic basketball actions	RGB-D Video Data	Yes
Shakya *et al.* [187]	two point, three point, mid range shots (success and failures separately classified)	RGB Video and optical flow data	Yes
Gu *et al.* [80]	3 broad categories: dribbling, passing, shooting; 26 fine-grained actions	RBG Video Data	Yes
Francia [68]	walk, no action, run, defense, dribble, ball in hand, pass, block, pick, shot	RGB Video Data	Yes
Parisot *et al.* [163]	player detection	RGB Video Data	Yes
De Vleeschouwer *et al.* [60]	Throw, Violation, Foul Player Exchange, Pass Rebound, Movement	7 cameras, RGB Video Data	Yes, upon request
Maksai *et al.* [133]	Trajectory estimation	RGB Data of various ball sports (basketball among others)	Yes*
Ramanthan *et al.* [169]	layups, free throw, 3 point, 2 point shots, slamdunk (success and failures separately classified)	RGB Video Data	Yes*
Tian *et al.* [206]	basketball tactics detection	RGB Video Data published by [236]	Yes

[60], mixed basketball activities like throwing, passing, or rebounding the ball, with detecting context-based activities like a player exchange, rule violation, or foul. Maksai *et al.* [133] estimated the trajectory of a ball in different sports, including basketball. Ramanathan *et al.* [169] focused on scoring activities, like performing layups, 3 and 2-point shots, free throws, and slamdunks. Although a large number of activity studies exist that explore sports data and basketball data, in particular, the number of publicly available benchmark datasets is significantly low. A fine-grained IMU-based sports dataset representing a single sport has become available only recently, following the publication of our Hang-Time HAR [93].

2.2.4. Deep Learning in Activity Recognition

Machine Learning for sensor-based Human Activity Recognition has a long tradition. Many published papers in the last two decades have proven its feasibility [180, 215, 231, 21, 127, 83, 159, 89]. While in the beginning most of the publications worked with classical Machine Learning approaches [215], [21], nowadays Deep Learning has replaced classical Machine Learning as the state-of-the-art learning algorithm [113], [221], because deep learning-based classifiers often outperform classical machine learning approaches. Latest since [83], IMU sensor signals are used as input to train neural networks. However, Deep Learning models have the disadvantage that their

success relies heavily on large amounts of data to be able to converge [39], [151].

During recent years, there have been notable advancements in the domain of deep learning applied to sensor data analysis. These developments include a range of techniques and methodologies that have significantly enhanced our ability to process and extract meaningful insights from sensor-generated information. Key trends in this field include the adoption of self-supervised learning [199], where a model learns representations or features from unlabeled data without explicit supervision; the utilization of few-shot learning approaches to generalize from limited annotated data [156]; the exploration of federated learning techniques for privacy-preserving and distributed model training [228, 192]; the integration of attention mechanisms to dynamically focus on the most salient time-intervals of sensor measurements [50]; the application of Temporal Convolutional Neural Networks for efficient sequential data analysis [150]; the adaptation of transformer networks for capturing long-range dependencies; the employment of Graph Neural Networks (GNNs) for modeling complex sensor networks [145, 120]; the effectiveness of transfer learning strategies to leverage pre-trained models [76, 95, 129]; the use of data augmentation to enhance model generalization [94, 212]; the fusion of multi-modal sensor data to provide comprehensive insights [239, 54]; the realization of real-time processing capabilities [100, 148]; the deployment of deep learning on edge devices for low-latency applications [8, 243]; and the advancement of explainable deep learning techniques to enhance model interpretability [64, 98]. Additionally, noteworthy progress has been made in model compression and energy efficiency [35], further enhancing the applicability and sustainability of deep learning approaches in sensor data analysis.

Chapter 5 specifically centers its attention on Transfer Learning and Data Augmentation techniques applied to sensor data. Consequently, the subsequent sections will place an emphasis on these two subfields.

Transfer Learning. Transfer Learning has become an increasingly important subtopic of Deep Learning in recent years. Therefore it was only a question of time until it was investigated whether this technology can be transferred to time-discrete sensor signals and thus also to Human Activity Recognition. The number of published papers in this discipline has increased rapidly, e.g. [97], [232], [63], [52], [146] or [118], especially in the last two years [88]. [146] showed a setup that I built upon and expanded with tests that artificially mapped the sensor's placement and orientation to each other, according to the results of [110] and [245]. Here it is shown, that only after the sensors have been brought into alignment, the classifier is achieving the best results. [76] showed, that cross-dataset transfer learning is possible if the source and target datasets are coming from the same domain. By using an architecture called MultiResNet [75], which transfers the data into the frequency domain and uses residual blocks they achieved promising results when transferring from Skoda Mini Checkpoint [237] to OPPORTUNITY [176], PAMAP2 [173] or JSI-FOS [74]. It seems like due to the transformation into frequency domain the trained filters are not class specific anymore and the orientation and location of the sensors axes lose their importance

for the success of Transfer Learning. **Our publication [95] primarily addresses inter- and intra-dataset sensor transfer, along with essential pre-processing steps aimed at enhancing the final model capabilities.**

Data Augmentation. Data Augmentation is one of the standard regularization techniques to prevent neural networks from overfitting [62] and in recent years, it has become an important focus in sensor-based Human Activity Recognition research. The idea to use synthesized data for training neural networks comes originally from computer vision, e.g. [101], [188]. Traditional transformations of images, e.g. scale, zoom, crop, or add noise to the data were adapted and transferred to time-series data by [212]. By applying these techniques, the original data gets slightly modified. In reverse, this also means that we never generate new and unique data. Another approach introduced by Ian Goodfellow [78] to augment data is to use a Generative Adversarial Network, like e.g. [244], [67]. Especially [67] is of importance, since I used this architecture as a baseline architecture, on which my system is built. The GAN published by Esteban et al. [67] consists of two neural networks. A generator model is used to augment data while a discriminator tries to distinguish between real and augmented data. These two models are training each other. As soon as the discriminator is no longer able to detect that the produced samples are synthesized, it is assumed that real-appearing time-series data is generated. An advantage of this architecture is that we generate new and therefore unique data, thus increasing not only the number but also the variability. **Our publication [94] examines how the selection of class instances for augmentation, followed by their inclusion into the dataset, impacts the overall performance of a deep learning model.**

Chapter 3

Towards Real-World User Studies for HAR

In the domain of wearable sensor technology, the pursuit of cost-effective and reliable data collection methodologies has emerged as a critical area of research. This holds the promise of not only enhancing data acquisition for individual researchers but also contributing valuable insights to the broader scientific community. The following chapter tries to answer important research questions focused on study recording techniques and participants' behavior during studies.

(a) **How can Data Sharing and Accessibility Be Maximized?**
In what ways can the data collected through wearable sensors be shared and made accessible to the wider research community?

(b) **Is Multisensor Synchronization Achievable through Cost-efficient Operations?**
Can synchronization points be accurately determined for multiple accelerometer signals recorded in parallel, using techniques such as correlation?

(c) **What is the Influence of Annotation Methods on Annotations' Quality and Quantity?**
How does the choice of annotation method impact the final quality and quantity of labeled data acquired through wearable sensors?

(d) **Which Annotation Method Best Suits Specific Research Use Cases?**
Which annotation methods align most effectively with specific research objectives and use cases?

(e) **Can the Hawthorne Effect and Behavior Changes be Detected through Sensor Data?**
- Do study participants alter their behavior when they are aware of being observed during experiments, and can this behavioral shift be detected through data collected from wrist-worn sensors?
- To what extent can deep learning classifiers identify and quantify such changes?

Section 3.1

Activate Data Recording System

[92] Hoelzemann, Alexander, et al.
Open-Source Data Collection for Activity Studies at Scale
May 2022, Part of the Smart Innovation, Systems and Technologies book series (SIST, volume 291)
https://doi.org/10.1007/978-981-19-0361-8_2

Portions of the original publication have been removed or edited for inclusion in this thesis. However, no changes were made that altered the results or conclusions presented in the original work.

Contributions:

- I designed and implemented the smartphone front-end, the backend, and the communication between both.
- Kristof Van Laerhoven guided this work and assisted in the methodologies. He developed the Activate firmware and the web-based control panel.

Activity studies range from detecting key indicators such as steps, active minutes, or sedentary bouts, to the recognition of physical activities such as specific fitness exercises. Such types of activity recognition rely on large amounts of data from multiple persons, especially with deep learning. However, current benchmark datasets rarely have more than a dozen participants. Once wearable devices are phased out, closed algorithms that operate on the sensor data are hard to reproduce and devices supply raw data. We present an open-source and cost-effective framework that can capture daily activities and routines, and which uses publicly available algorithms while avoiding any device-specific implementations. In a feasibility study, we were able to test our system in production mode. For this purpose, we distributed the Bangle.js 1 smartwatch as well as our app to 12 study participants, who started the watches at a time of individual choice every day. The collected data was then transferred to the server at the end of each day.

3.1.1. Introduction

Many types of studies focus on capturing activity data from human study participants. We can distinguish these types of studies based on the measurement devices and sensors used, the carrying position of the sensors, and the domain of the data. The types of devices used go hand in hand with the sensor technology used. For example, sensors worn on the wrist offer the possibility of recording the heart rate via PPG sensors, the skin temperature with a thermometer, and the movements with an accelerometer, gyroscope, and magnetometer. Studies in which smartphones are mainly used to record data do not usually offer this supplementary sensor technology. Since the

devices are not worn directly on the skin, the data is often limited to basic IMU sensors. In contrast, the carrying position of the sensors goes hand in hand with the specific domain of the recorded data. The sensor technology used for medical datasets is often worn on different body positions than sensor technology used for activity recognition. As previous studies have shown, for many activities, it is often sufficient to wear the sensors only at key positions such as the wrist [110], [136]. In the medical environment, however, more complex sensors and different wearing positions are often required [107], [10]. Empirical studies for which activity plays a crucial role use indicators such as steps taken, sedentary periods, activity counts, or detected physical exercises, which often originate from closed-source algorithms. This tends to lock studies to particular devices and makes the use of other devices or comparisons difficult. Restricting studies to particular commercial wearables that also record raw inertial data has the effect that large-scale studies are only possible if the project has a high budget that allows the purchase of commercial hardware and software. In this section, we present the ActiVatE_prevention system, see Figure 3.1, which is based exclusively on open-source components, logs raw inertial data, and also offers subjects a similar wearing comfort as commercially manufactured products. We argue that it, therefore, lends itself well to the capturing of multiple users simultaneously for activity studies, while being an open-source, replicable, and low-cost approach.

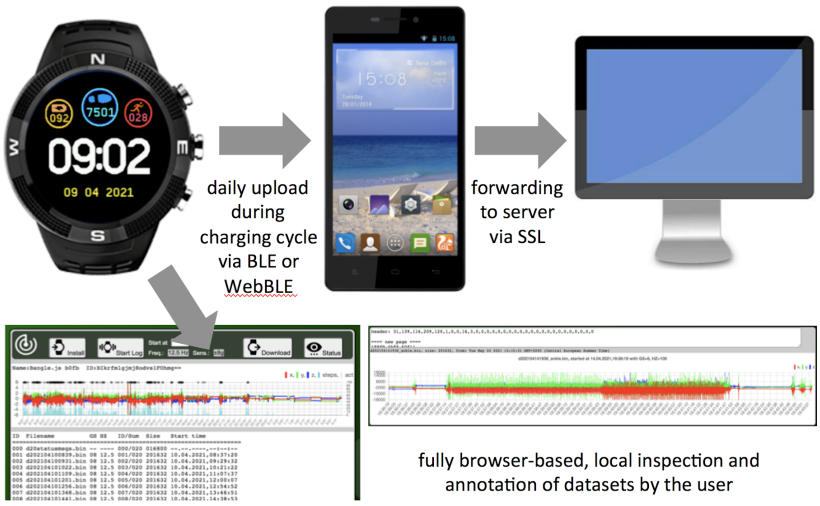


Figure 3.1 Our system relies on an open-source smartwatch [226] with custom firmware, smartphone apps, and a server-side database to collect all data centrally. For participants without a smartphone or in studies where users need to inspect their data or manually forward their data, a web-based suite (bottom) retrieves the data through a (Web)-Bluetooth Low Energy (BLE) connection. The raw sensor data is frequently streamed from the smartwatch either to a nearby computer via a web-based control panel, or via the user's smartphone to a dedicated server.

3.1.2. Our Proposed Approach

The design of our open source system is shown in Figures 3.2 and 3.3. The operating system is installed once on the Bangle.js via Web-BLE and the apps are downloadable via the Apple AppStore and the Google Play Store. The app forwards the data from the smartwatch to the central server. The user interface of the app is kept simple, the users can only select their daily activity goals and retrieve their daily activity statistics.

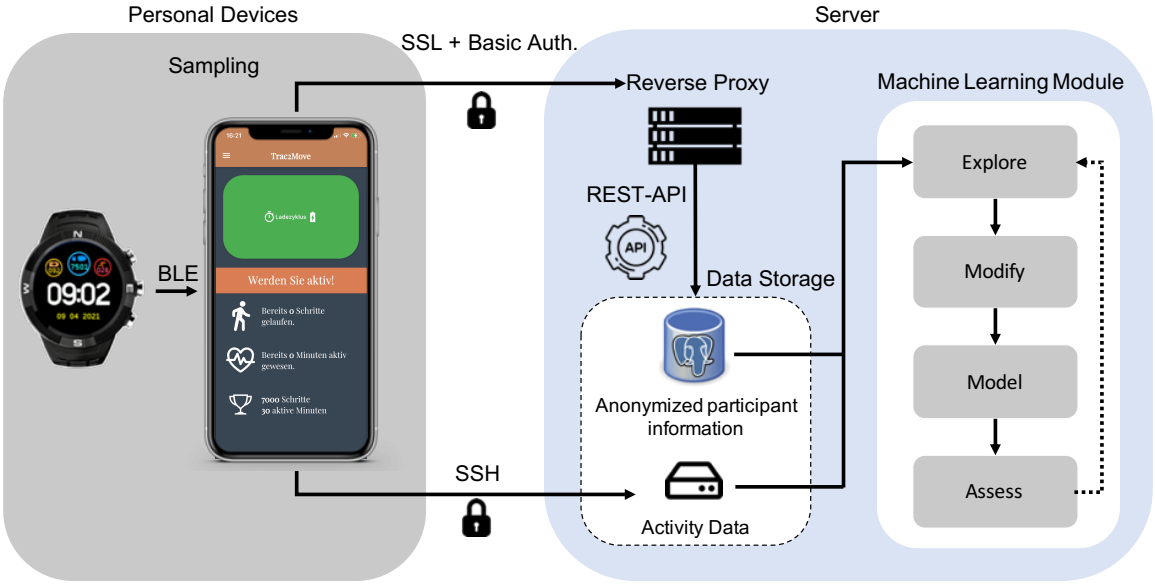


Figure 3.2 Open source client-server architecture for recording human activity data. The data is recorded by the Bangle.js smartwatch and is sent to the server daily with our app. Anonymized participant information is sent to the server via a reverse proxy that implements a Secure Sockets Layer (SSL) + Basic Authentication. This reverse proxy communicates via a REST-API with the Postgres Structured Query Language (SQL) database. The system is designed following the Sample, Explore, Modify, Model, and Assess (SEMMA) data process model [186]. (1) Sampling, (2) Explore (3) Modify (4) Model (5) Asses. The model itself can be seen as a cycle.

The sequence diagram (Figure 3.3) depicts the communication in between the architecture elements. We recorded the execution time for every communication step, which is added to the diagram. On average, it takes 185 seconds to send one file (approx. 200KB and 1 hour of data) from the watch via BLE to the smart device. After ∅ 45 minutes, the complete daily data is sent from the smartwatch to the server using our Representational State Transfer (REST) - Application Programming Interface (API). **Smartwatch.** To date, few open-source smartwatch designs allow algorithms for detecting activities, from basic ones such as steps, sedentary bouts, and active minutes, to recognition of particular exercise repetitions, to be transparently implemented on a device with integrated inertial sensors. We used the Bangle.js [226] as an affordable, around $50 USD, low-power system that is equipped with a Nordic

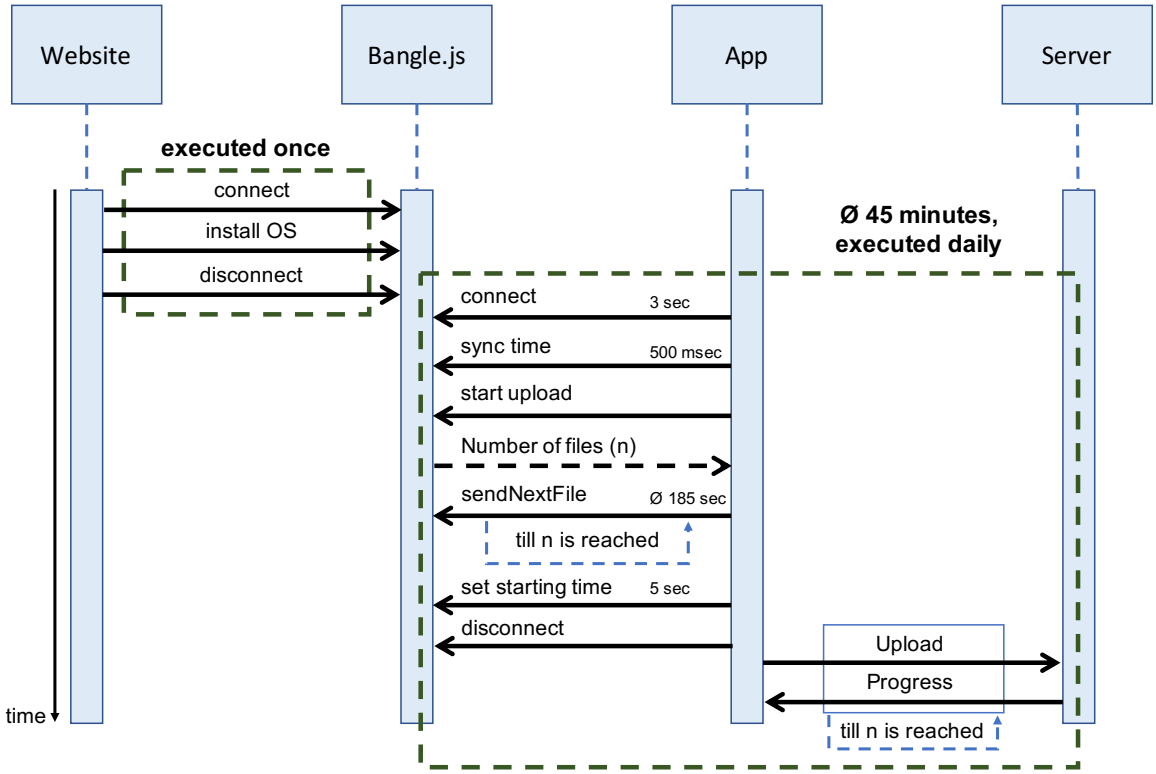


Figure 3.3 Activate System Sequence Diagram: The operating system is installed from our web tool via Web-BLE on the Bangle.js. This needs to be executed once. The communication between Bangle.js and the app occurs on a daily basis. The procedure needs ∅ 45 minutes for a full day of recording (14 hours of active time) and ∅ 185 seconds for sending one file from the watch to the smartphone. The upload to the server is executed when all files are transferred to the smartphone.

64MHz nRF52832 ARM Cortex-M4 processor, inertial sensors, a PPG sensor, sufficient internal memory, and an internal BLE module. Our firmware on this open-source platform is capable of storing the sensors' raw data over a full day, and integrating recognition algorithms – currently for steps, active minutes, and exercise intensities – locally on the watch. Users are expected to start the data upload process once a day, either through the web-based platform or automatically through their smartphone or tablet app.

Since the logging of activity data requires sampling rates from 10Hz up to as high as 100Hz, depending on the activity, the recording of raw inertial data is rarely implemented in a way where local recordings are routinely synchronized and uploaded to a server. The local storage for a day's worth of inertial data and the energy footprint for sending this data tends to be substantial [27]. Instead, the early pre-processing of inertial data in the aforementioned detected features (steps, active minutes, etc) takes place on the wearable devices and usually solely these aggregated values are stored.

Detected activity-related concepts such as Active Minutes [104] have been deployed locally on the Bangle.js smartwatch and are uploaded together with the raw sensor data

to the server through the smartphone app (or via the browser-based tool suite) daily. We designed to fully use the watch's 4Mb flash memory to losslessly compress 16 bit, 12.5 Hz inertial data at +/-8g, along with other data such as the skin temperature and heart rate.

iOS and Android App. The Activate client is implemented using Flutter. Therefore, we can design and implement clients for the two major operating systems, iOS and Android, at once. However, minor code changes are necessary to solve operating system-specific issues, especially with regard to the BLE connection.

The interface consists of three main views and is displayed in German language. It was designed to encourage diabetes patients to perform more physical activities in their daily lives. Beyond the recording of raw inertial data, it is planned for the near future to expand this open-source app to be able to annotate and detect an arbitrary number of activities as well. When the app starts, the participant is taken to the home screen, (1) in Figure 3.4. Here, the user interface visualizes an overview of the day's accumulated number of steps taken and active minutes. When pressing the green button, the study participant saves the data on the server and sets the starting time for the following measurement (typically the next day). During the first start of the app, an anonymized user account is created and saved in a Postgres Structured Query Language (SQL) database. On the second screen in Figure 3.4, the user can

Figure 3.4 The smartphone's user interface: (1) Home Screen, (2) Setting daily activity goals, e.g. Daily Steps (Tägliche Schritte) and daily Active Minutes (Aktive Minuten), (3) Graphical overview of daily activities: Daily Steps (Tägliche Schritte), Active Minutes (Aktive Minuten), divided into three intensities - low, moderate and vigorous (niedrige, mittlere, hohe Intensität).

set their personal goals for the day within its limits. Screen (3) in Figure 3.4 gives a graphical overview of the daily metrics and shows, besides the total number of steps and active minutes, the active minutes sorted by their intensities.

Server. The server communicates with the client via two channels, Figure 3.2. Private information about the study participants, such as gender or age, and the confirmation of the consent form are sent via Secure Sockets Layer (SSL) and Basic

Authentication to a reverse proxy which then sends the information to the database via localhost. The information is stored in an anonymous form. The recorded activity data, as well as daily steps and active minutes, are sent via SSH to the server and stored in binary files with delta compression. The activity data can then be processed and modeled by machine learning algorithms, e.g. Sample, Explore, Modify, Model, and Assess (SEMMA).

Browser-based data analysis. The smartphone or tablet app and server software described above can be complemented with a local analysis and annotation tool that can be used by the study participants. This requires users to simply visit a website that can connect to the watch through WebBluetooth Low Energy (BLE) and download the watch's data locally on the computer for further inspection or manually upload to our study server, through users' computers without the need to install software.

3.1.3. Performance Analysis

Since our software is distributed between apps that are available as a web-based software suit or downloadable in Apple's App Store and Android's Play Store, the deployment of our system is straightforward. We gave the Bangle.js smartwatches to 12 geographically distributed study participants and recorded compliance, comfort rating, and reliability performance measures for our presented approach to illustrate the feasibility of our approach and report our findings below.

We analyzed recordings from participants over a window of five days and decided to let them choose how many hours they recorded by letting them start and stop the smartwatch with the app at a time of their choice. This is important because of the age group and the profession of the subject, which entails certain active and inactive, as well as sleep and wake cycles [66].

During the feasibility study, we focused on detecting basic activity concepts such as steps as well as the active minutes divided into three subclasses, low, moderate, and vigorous intensity. The participants wore the smartwatch for an average of 12 hours per day. In total, we collected approx. 29 MB (12*202KB*12 participants) of raw compressed data. Basic activity classes are already recognized on the watch without machine learning. However, since Bangle.js has Tensorflow-Lite already implemented on the hardware, there is an opportunity to deploy a pre-trained neural network or machine learning classifier on the watch in the future. A recent article [132] demonstrates how to implement this for gesture recognition.

We can demonstrate through our experiment that the system we have designed can be used for data recordings in-the-wild without the subject being biased by the technology worn since the smartwatch is a commercially designed product and looks and feels like a normal watch. Due to the 4 Mb memory limitation of Bangle.js, we limit the inertial measurements to a 12.5 Hz sampling rate so that a full 24-hour day can still be recorded in one cycle. We consider this sampling rate acceptable since activity detection is still possible at such a low sampling rate. Furthermore, the signal can be interpolated as part of the machine learning preprocessing or the sampling

rate can be increased at the cost of shorter recordings (a 100 Hz recorded data set corresponds to about 3 hours).

Occasionally, data uploads are hampered because of problems with a reliable Internet connection and the Bluetooth connection between Bangle.js and the app in particular. The communication flow as depicted in figure 3.3 has therefore been developed for stability and has built-in recovery mechanisms that guarantee that individual files are uploaded reliably. The current version is therefore characterized by high reliability and accessibility, but also relatively long upload times (around 45 minutes on average for a full day's data set). However, this seems acceptable, as the download process has been integrated with charging the smartphone and Bangle.js smartwatch in the nightly "charging cycle".

Bangle.js 1 Wearing Comfort. In addition to the feasibility study of our open-source architecture's ability to accommodate data over multiple users and in a distributed manner, we decided to investigate Bangle.js in terms of its comfort of use. We consider this to be important since the success of a study is directly dependent on the acceptance of a device. We use the Comfort Rating Scale (CRS), a questionnaire-based method proposed by Knight et al. [109], as a well-known and state-of-the-art method to evaluate the wearing comfort of wearable devices in particular. The Bangle.js smartwatch was rated (as Figure 3.5 shows) overall as comfortable to wear without restricting its users. However, users can feel the device

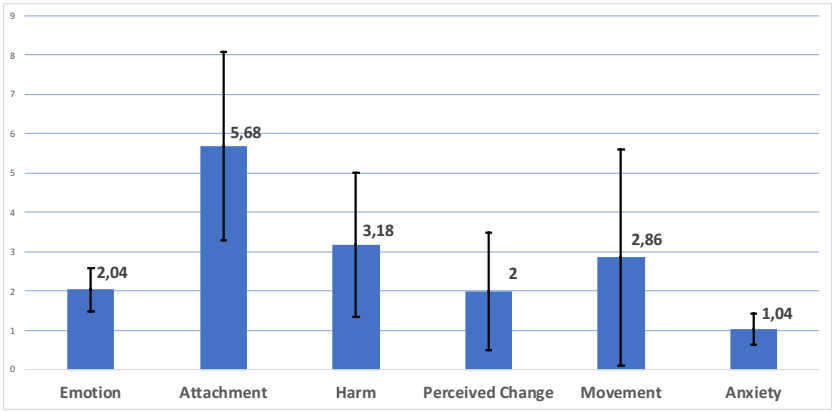


Figure 3.5 CRS result means and standard deviation. Emotion: 2.04, 0.56; Attachment: 5.68, 2.39; Harm: 3.18, 1.84; Perceived Change: 2, 1.49; Movement: 2.86, 2.75; Anxiety: 1.04, 0.39.

on their wrist due to its larger size (5 x 5 x 1.7 cm case) and weight. The device is heavier and more bulky than most commercial wrist-worn products, which may lead to slightly negative wearing comfort and perhaps more difficult acceptance in larger future studies. We consider this an acceptable trade-off, as only one person in the study reported that the watch had a strong negative emotional impact on them and that they would have liked to take it off.

3.1.4. Conclusions

The use of low-cost and open-source systems is essential for future machine learning applications. Only through the development and use of such systems, it will be possible

to generate the required amount of data to train a neural network to be used in a real-world context in a generalized way. Many publications show new and exciting methods in dealing with human activity data, however, these methods are always evaluated on the same datasets mentioned before. This creates a bias in our scientific domain, which can only be eliminated by publicly available, understandable, and reusable implementations for data collection.

The already available open-source platforms and systems presented in chapter 2.2 are either smartphone-based or smart-home-based solutions. Smartwatch-based solutions are mostly prototypes, which are not meant to be distributed in scale and are not open-source. Due to its open-source architecture, the use of the Bangle.js wristwatch combines the advantages of a product while having an open architecture that is fully documented. Our custom operating system as well as the client-server architecture can serve as a starting point that can later be modified or further developed accordingly. Due to the low purchase price, the device can be used in projects with a smaller budget or need of a larger group of users. In contrast to a self-developed prototype, where wearing comfort is often not the main interest, the Bangle.js was confirmed to offer a high acceptance by study participants in our study using the comfort rating scale (CRS). We argue that this aspect also contributes to the long-term success of a scientific study and the scope, quality, continuity, and reliability of the produced dataset. Commercial products tend to not open the algorithms used and do not give researchers the same insights into recorded data as a fully open-source implementation does. Therefore, we made the source code of the smartphone app as well as the smartwatch operating system available for download and inspection under the MIT license, to encourage other researchers to replicate and improve on our approach:
`https://github.com/ahoelzemann/activateFlutter`,
`https://github.com/kristofvl/BangleApps/blob/master/apps/activate/app.js`

Section 3.2

Comparing Annotation Methods

[96] Hoelzemann, Alexander, and Van Laerhoven, Kristof.
A Matter of Annotation: An Empirical Study on In Situ and Self-Recall Activity Annotations from Wearable Sensors
January 2024, Currently Under Review for Frontiers in Computer Science
Manuscript Number: 1379788
A preprint was submitted to the arxiv repository of the Cornell University
https://doi.org/10.48550/arXiv.2305.08752

Portions of the original publication have been removed or edited for inclusion in this thesis. However, no changes were made that altered the results or conclusions presented in the original work.

Contributions:

- The study was designed by both authors, but executed and analyzed by me.
- Kristof Van Laerhoven guided this work and assisted in the methodologies.

Research into the detection of human activities from wearable sensors is a highly active field, benefiting numerous applications, from ambulatory monitoring of healthcare patients via fitness coaching to streamlining manual work processes. We present an empirical study that compares 4 different commonly used annotation methods utilized in user studies that focus on in-the-wild data. These methods can be grouped into user-driven, in situ annotations - which are performed before or during the activity is recorded - and recall methods - where participants annotate their data in hindsight at the end of the day. Our study illustrates that different labeling methodologies directly impact the annotations' quality, as well as the capabilities of a deep learning classifier trained with the data. We noticed that in situ methods produce less but more precise labels than recall methods. Furthermore, we combined an activity diary with a visualization tool that enables the participant to inspect and label their activity data. Due to the introduction of such a tool were able to decrease missing annotations and increase the annotation consistency, and therefore the F1-Score of the deep learning model by up to 8% (ranging between 82.1 and 90.4 % F1-Score). Furthermore, we discuss the advantages and disadvantages of the methods compared in our study, the biases they could introduce, and the consequences of their usage on human activity recognition studies as well as possible solutions.

3.2.1. Introduction

Sensor-based activity recognition is one of the research fields of Pervasive Computing developed with enormous speed and success by industry and science and influencing

medicine, sports, industry, and therefore the daily lives of many people. However, current available smart devices are mostly capable of detecting periodic activities like simple locomotions. In order to recognize more complex activities a multimodal sensor input, such as [176], and more complex recognition models are needed. Many of the published datasets are made in controlled laboratory environments. Such data does not have the same characteristics and patterns as data recorded in-the-wild. Data that belongs to similar classes but is recorded in an uncontrolled versus controlled environment can differ significantly since it contains more contextual information [140]. Furthermore, study participants tend to control their movements more while being monitored [69]. The recording of long-term and real-world data is a tedious, time-consuming, and therefore a non-trivial task. Researchers have various motivations to record such datasets but the technical hurdles are still high and problems during the annotation process occur regularly. With regards to Human Activity Recognition (HAR), recording a long-term dataset always presents the researcher with the problem of developing a methodology that, on one hand, allows precise labels and, on the other hand, does not unnecessarily burden or disturb the study participants. Relying only on self-recall methods, like writing an activity diary, e.g. [241], can result in imprecise time indications that do not necessarily correspond to the actual time periods of an activity. Such incorrectly or noisy labeled data [152] leads to a trained model that is less capable of detecting activities reliably, due to unwanted temporal dependencies learned by wrongly annotated patterns [33].

We can see an emerging spotlight on real-world and long-term activity recognition and think that it will be one of the main research challenges that need to be put more in focus to overcome current limitations and be capable of recognizing complex day-to-day activities. Such datasets, rely heavily on self-recall methods or using additional apps to track movements and set labels either automatically [9] or with the manual selection of a label [56]. Due to these hurdles, many researchers prefer to work with datasets from controlled over uncontrolled environments. As a consequence, only a very limited number of in-the-wild datasets have been published until now.
Contribution: Our study focuses on the evaluation of 4 different annotation methods for labeling data in-the-wild: ① In situ *(lat. on site or in position)* with a button on a smartwatch, ② in situ with the app Strava [1] (an app that is available for iOS and Android smartphones), ③ pure self-recall (writing an activity diary at the end of the day), and ④ *time-series* assisted self-recall with the MAD-GUI [157], which displays the sensor data visually and allows to annotate it interactively. Our study was conducted with 11 participants, 10 males, and 1 female, over 2 weeks. Participants wore a Bangle.js Version 1[2] smartwatch on their preferred hand, used Strava, and completed self-recall annotations every evening. In the first week, the participants were asked to write an activity diary at the end of the day without any helping material and additionally using two user-initiated methods (*in situ button* and *in situ app*) to manually set labels at the start and beginning of each activity. In the second week, the

[1] `https://www.strava.com/`
[2] `https://www.espruino.com/banglejs`

participants were given an additional visualization of the sensor data with an adapted version of the MAD-GUI annotation tool. With the help of this, participants then were instructed to label their data in hindsight with the activity diary as a mnemonic aid. Given labels from both weeks were compared to each other regarding the quality through visual inspection and statistical analysis with regard to the consistency and quantity of missing annotations across labeling methods. Furthermore, we used a Shallow-DeepConv(LSTM) architecture, see Bock *et al.* [35] and Ordoñez *et al.* [159], and trained models with a leave-one-day-out cross-validation method of 6 previously selected subjects and each annotation method.

Impact: Annotating data, especially in real-world environments, is still very difficult and tedious. Labeling such data is always a trade-off between accuracy and workload for the study participants or annotators. We raise awareness among researchers to put more effort into exploring new annotation methods to overcome this issue. Our study shows that different labeling methodologies have a direct impact on the quality of annotations. With the deep learning analysis, we prove that this impacts the model capabilities directly. Therefore, we consider the evaluation of frequently used annotation methods for real-world and long-term studies to be crucial to give decision-makers of future studies a better base on which they can choose the annotation methodology for their study in a targeted way.

3.2.2. Study Setup

Our study is conducted with 11 participants, from which 10 are male and 1 is female. The participants are between 25 and 45 years old. Out of 11 participants, 6 are researchers in the field of signal processing and are used to read and work with sensor data. Participants were selected among acquaintances and colleagues. The study was conducted over 2 weeks while participants wore an open-source smartwatch on their wrist of choice. During the two-week study, the participants were instructed to use 4 different labeling methods in parallel, see Figure 3.6. In the first week they were asked to use the ① *in situ button*, ② *in situ app*, and ③ *pure self-recall* methods. At the beginning of the 2nd week, we expanded the number of annotation methods with the ④ *time-series recall*. This annotation method combines the activity diary with a graphical visualization of the participants' daily data.

① The Bangle.js smartwatch has 3 mechanical buttons on the right side of the case. These buttons are programmed to record the number of consecutive button presses per minute. The total number of button presses is stored with the given timestamp and can therefore be used to mark the beginning and end of an activity in the time-series.

② In addition, the participants were asked to track their activities with the smartphone app Strava. Strava is an activity tracker that is available for Android and iOS and freely downloadable from the app stores. The user can choose from a variety of predefined labels and start recording. Recording an activity starts a timer that runs until the user stops it. The time as well as the Global Positioning System (GPS) position of the user during the activity is tracked and saved locally.

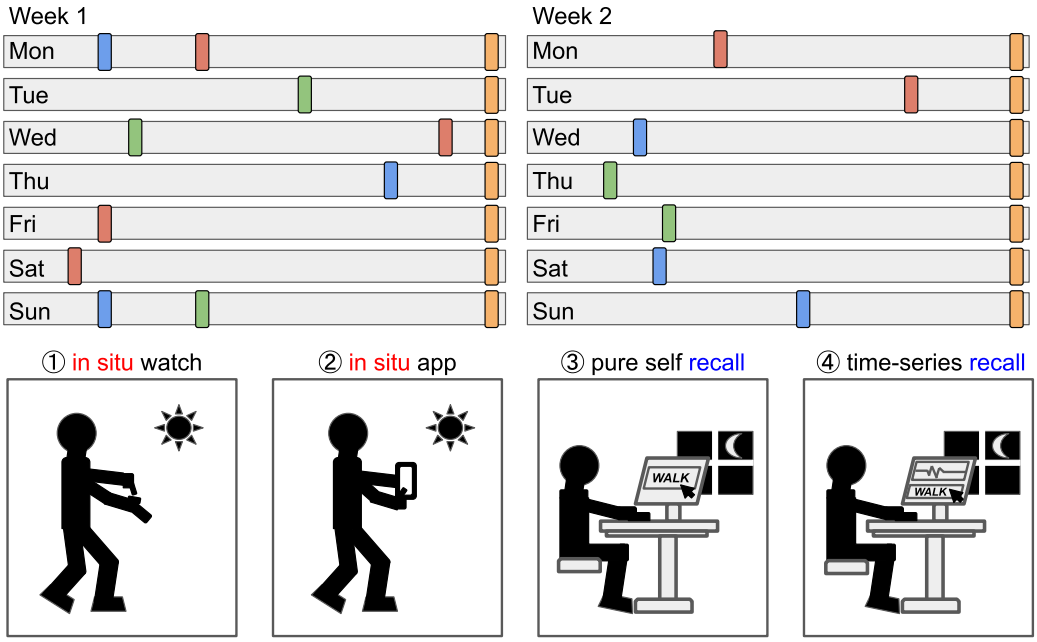


Figure 3.6 The study participants collected data for 14 days in total and annotated the data with 4 different methods: Labeling ① in situ with a mechanical button, ② in situ with an app, ③ by writing a pure self-recall diary and ④ writing a self-recall diary assisted by visualization of their time-series data.

③ The *pure self-recall* methods consist of writing an activity diary on a daily basis at the end of the day. The participants were explicitly told that they should only write down the activities that they still remember 2 hours after the measurement stopped.

④ The *time-series recall* method can be seen as a combination of an activity diary and a graphical representation of the raw sensor data. For visualization and labeling purposes, we provided the participants with an adapted version of the MAD-GUI. The Graphical User Interface (GUI) was published by Ollenschläger *et al.* [157] in 2022 and is a generic open-source Python package. Therefore, it can be integrated into one's project. Our adaptions to the package are available for download from a GitHub repository[3]. It contains changes to the data loader, the definition of available labels, and color settings for displaying the 3D raw data.

Annotation Guidelines: The participants were briefed to note daily returning activities (sports or activities of daily living) that are performed longer than 10 minutes roughly 2 hours after the recording stops. The name of the activity was chosen individually by the participant. Participants decided individually at what time of the day the recording would start on the next day. Each of these annotation methods represents a layer of annotation that is used for the visual, statistical, and deep learning evaluation. Figure 3.6 illustrates the overall concept.

[3] `https://github.com/ahoelzemann/mad-gui-adaptions/`

3.2.3. Statistical Analysis

The labels were statistically analyzed based on their consistency using the Cohen κ score as well as the number of missing annotations across all methods. The Cohen κ score describes the agreement between two annotation methods, which is defined as follows $\kappa = (p_0 - p_e)/(1 - p_e)$ (see [16] and [172]). Where p_0 is the observed agreement ratio and p_e is the expected agreement if both annotators assign labels randomly. The score shows how uniform two different annotators labeled the same data. For calculation purposes, an implementation provided by Scikit-Learn [183], was used. Furthermore, missing annotations across methods are measured as the percentage of missing or incomplete annotations. The annotations of all methods were first compared with each other and matched based on the given time indications. Annotations that could not be assigned or were missing were marked accordingly and are the base for calculating this indicator, Figure 3.9 visualizes this. We used a similar representation as [40] to visualize the matches among labeling methods. In this study, the authors compared genome annotations labeled by different annotators with regard to their error scores between different annotators.

3.2.4. Effects on Deep Learning Performance

The deep learning analyses are performed using the DeepConvLSTM architecture [159] which is based on a Keras implementation of [95]. We did not perform hyperparameter tuning because it would involve a considerable amount of additional workload, since we trained 64 models independently during the evaluation. We therefore decided to opt out of the architecture with regards to efficiency rather than optimal classification results. Additionally, we don't expect that the actual experiment - evaluating different annotation methods - would benefit from hyperparameter tuning or gain any significant information and insights. Instead, we use the default hyperparameters provided by the authors. These are depicted in the Figure 3.7. Furthermore, we reduce the number of LSTM layers to one and instead increase the number of hidden units of the only LSTM layer to 512. According to [35], this modification decreases the runtime up to 48 % compared to a two-layered DeepConvLSTM while significantly increasing the overall classification performance on 4/5 publicly available datasets: [174], [198], [176], [181], [195]. LSTM-Layers in general are important if the dataset contains sporadic activities [33]. However, our dataset does not and our evaluation aims to identify long periods of periodic activities, like walking or running. For this reason, we can conclude that additional LSTM layers are not needed. The implementation of [95] incorporates BatchNormalization layers after each Convolutional layer, as well as MaxPooling for the transition between the final convolutional block and the LSTM layer, and a Dropout layer before classification. Each Convolutional layer employs a Rectified Linear Unit (ReLU) activation function. The inclusion of the BatchNormalization layers serves to accelerate training and mitigate the detrimental effects of internal covariate shift, as discussed further in [102].

Before training the neural network we apply two preprocessing steps to the data.

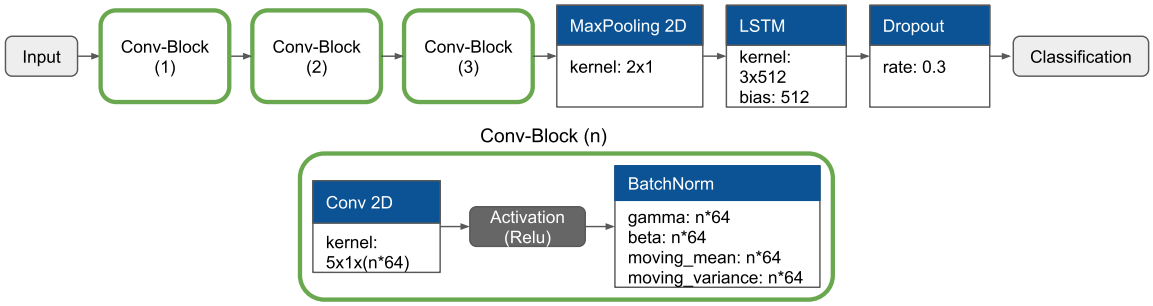


Figure 3.7 The architecture consists of an **Input** Layer with the kernel-size 10 (window_size) x 10 (filter_length) x 3 (channels). The data is passed into 3 concatenated **convolutional blocks**, followed by a **MaxPooling** (kernel 2x1) where 50% of the data is filtered. The convolutional block consists of a convolutional layer with a variable kernel size of 5x1x(n*64) following a Rectified Linear Unit (ReLU) activation function and a BatchNorm-Layer. We decided to use a single LSTM-Layer with the size of 512 units, as mentioned by [35], which is followed by a Dropout-Layer that filters 30% of randomly selected samples of the window.

These are an upsampling to a constant 25 Hz due to small deviations in the device sampling rate as well as a rescaling of the accelerometer data between -1 and 1.

Leave-One-Day-Out Cross-Validation: Figure 3.8 shows the train- and test-setting for the deep learning model. For every study participant and every week, a personalized model has been trained and tested. We adapted the leave-one-subject-out strategy to fit our needs. Instead of using one participant for testing, we used one day of the week for testing and trained on every other day. The dataset consists of a major *void*-class and a small number of samples per activity class and participant. To counteract an above-average large *void*-class, we trained our model with balanced class weights.

Not limiting the participants in their choice of daily performed activities as well as well as not specifying predefined activity labels resulted in very unique sets of activities per study participant. Due to these circumstances, we cannot expect that it is possible to train a model that is capable of generalizing across participants and days. Every participant comes with their specific patterns to perform an activity. Furthermore, due to the in-the-wild recording setup, the intra-class differences [45] for comparatively simple activities, *walking* or *running* can be significant. The impact of different labeling methods is therefore expected to be more present, and hence more visible, in a personalized than it would in a generalized model.

Postprocessing & Classification: We classified the data based on a sliding window with a length of 2 seconds (50 samples). However, we are looking for longer periods of reoccurring activities and decided therefore to apply a jumping window of 5 minutes that applies a majority vote to the given time period. The activity with the most instances in those 5 minutes is set for the whole window.

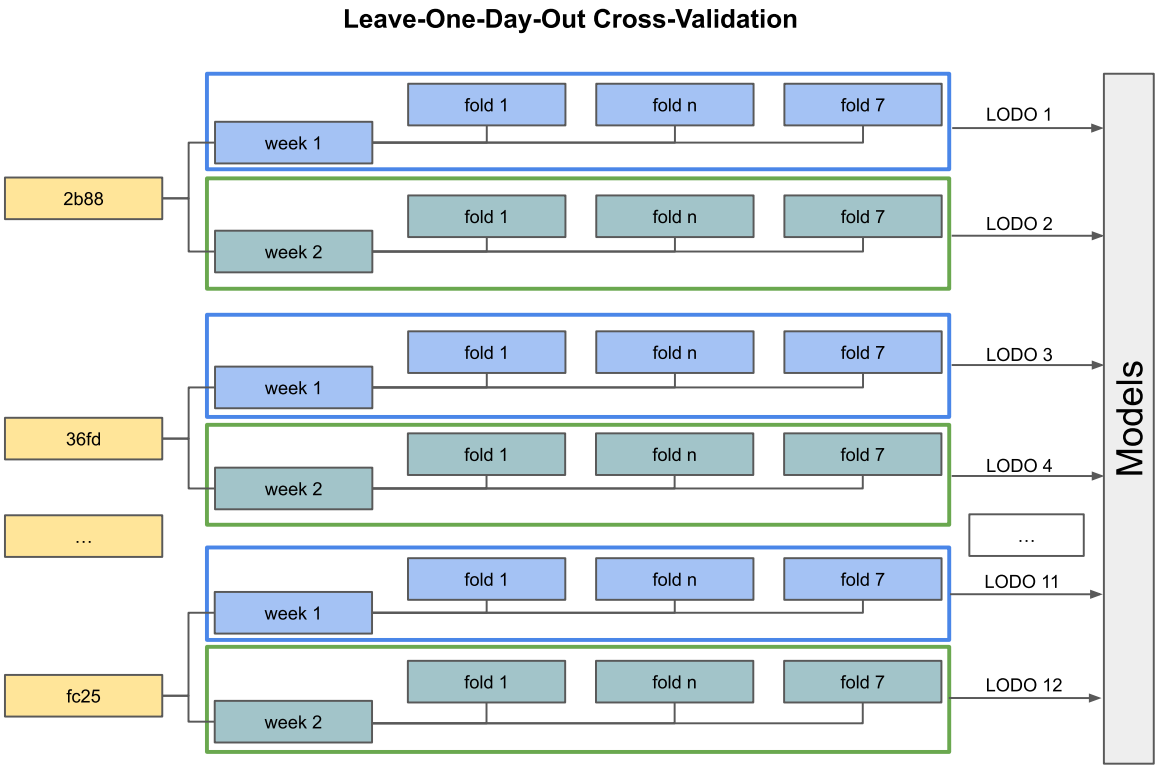


Figure 3.8 Leave-One-Day-Out Cross Validation. The models are personally trained for every participant and are not intended to generalize across all study participants. Instead, a generalization across all days of one week is desired.

3.2.5. Results

Our participants were asked to annotate activities carried out in their daily lives performed for more than 10 min. We didn't limit the participants to a predefined set of classes. They decided independently which labels they would like to set for certain periods. After normalizing the labels, e.g. changing "going for a walk" to "walking" etc., the set of labels as given by the participants contains the following 23 (22 + void) different labels: *laying, sitting, walking, running, cycling, bus_driving, car_driving, vacuum_cleaning, laundry, cooking, eating, shopping, showering, yoga, sport, playing_games, desk_work, guitar_playing, gardening, table_tennis, badminton, horse_riding.* Every sample that wasn't specifically labeled is classified as *void*.

Missing Annotations and Consistency Across Methods. Missing Annotations and the consistency of labels set over the course of one week varied greatly depending on the study participant. However, tendencies with regard to specific methods are observable.

Method ①, pressing the situ button on the smartwatch's case, was not consistently used by every participant. Furthermore, this method carries the risk that either setting one of the two markers (start or end) is forgotten. An annotation where one marker is missing becomes therefore obsolete. The app-assisted annotation method ②, for

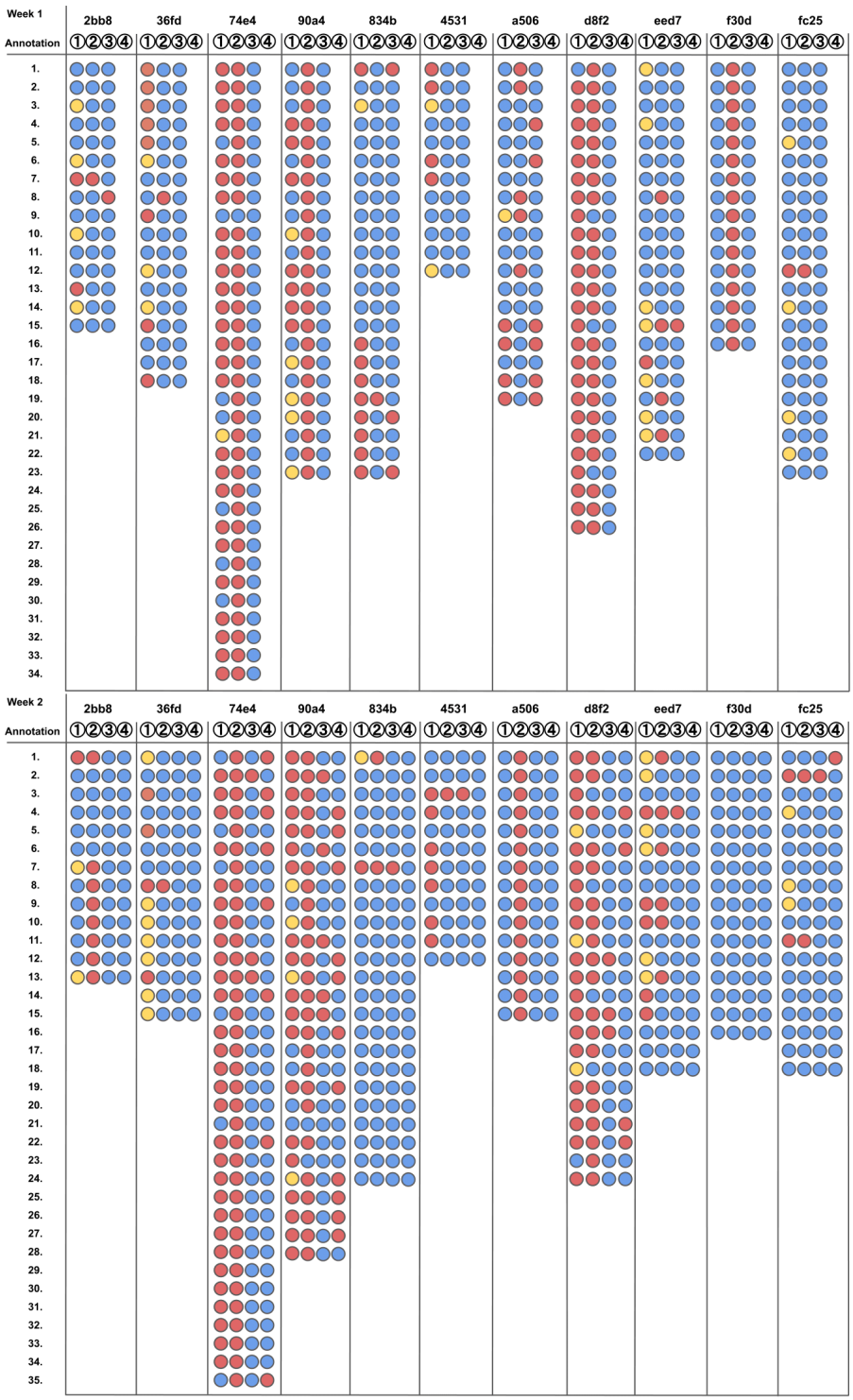


Figure 3.9 Missing annotations across all study participants and both weeks. The Y-axis shows the total number of annotations of one specific participant for the corresponding week. The color codes are as follows: ● Annotation is missing, ● Annotation is partially missing (start or stop time), ● Annotation is complete. The figure is inspired by [40], Figure 1.

which we used the app Strava, is well accepted among the participants who agreed with using third-party software. However, 4 participants, namely *74e4*, *90a4*, *d8f2* and *f30d* did not use the app continuously or refused to use it completely due to concerns regarding their private data. Strava is a commercial app, that is freely available for download on the app stores, but it collects certain users' metadata. To label a time period with Strava, the participant needs to (1) take the smartphone, (2) open the app, (3) start a timer, set a label, and (4) end the timer. This procedure contains significantly more steps than other methods. Therefore, the average value of missing annotations results in 46.40% (week 1) and 56.79% (week 2). One participant found the annotation process in general very tedious and therefore dropped out of the study. These data have been excluded from the dataset and the evaluation. Method ③ *pure self-recall*, writing an activity diary, got well accepted by every participant. As Figure 3.9 shows and the results in Table 3.1 proof, it is overall the most complete annotation method with an average amount of missing annotations of 4.30 % for the first and 8.14 % for the second week. By introducing the MAD-GUI, participants were able to

Table 3.1 Missing annotations across all labeling methods (in %) of both weeks. The columns contain the subject-ID of all participants. The last column shows the average percentage of missing annotations across every labeling method, for all participants.

Week 1												
Subject	**2b88**	**36fd**	**74e4**	**90a4**	**834b**	**4531**	**a506**	**d8f2**	**eed7**	**f30d**	**fc25**	**Avg.**
① ***in situ button***	40	70.59	79.41	52.18	36.37	50	26.32	96.15	45.46	0	26.81	40.95
② ***in situ app***	13.30	5.89	97.06	100	5.00	0	36.84	92.30	22.73	100	4.35	43.40
③ ***pure self-recall***	6.67	0	0	0	4.55	0	31.58	0	4.55	0	0	4.30
Week 2												
① ***in situ button***	23.08	73.33	92.00	82.14	8.33	76.47	0	95.33	61.11	0	27.78	49.05
② ***in situ app***	61.54	6.67	100	89.29	8.33	35.30	100	79.16	33.33	100	11.11	56.79
③ ***pure self-recall***	0	0	8.57	17.88	4.17	35.30	0	12.50	5.56	0	5.56	8.14
④ ***time-series recall***	0	0	22.88	39.29	0	0	0	16.67	0	0	5.56	7.67

inspect their daily data, get insights into what patterns of specific classes look like, and label them interactively. With an average amount of missing annotations of 7.67%, this method became the most complete during the second study week. Table 3.2 shows the resulting Cohen κ scores. Due to the constraint that only one labeling method can be compared to a second one and since, according to Table 3.1, the most consistent annotation methods are ③ *pure self-recall* and ④ *time-series recall*, we used these methods as our baseline and compared them with every other method used in the study. The second column indicates the comparison direction. The abbreviations used in this column are defined as follows: (③ C/W ①) *pure self-recall* compared with *in situ button*, (③ C/W ②) *pure self-recall* compared with *in situ app* and (③ C/W ④) *pure self-recall* compared with *time-series recall*. The direction (④ C/W ③) is not explicitly included since Cohens κ is bidirectional and both directions result in the same score. The score indicates how similar two annotators, or in our study labeling methods, are to each other. The resulting score is a decimal value between -1.0 and

Table 3.2 Average similarity between annotation methods according to the Cohan κ score for both study weeks. The columns are ordered subject-wise. The last column shows the average across all participants for one study week. The Direction column indicates in which the direction the Cohan κ score is calculated and needs to be interpreted as follows: ① *in situ button*, ② *in situ app*, ③ *pure self-recall*, ④ *time-series recall*, (C/W) compared with.

Week, Day	Direction	2b88	36fd	4531	74e4	834b	90a4	a506	d8f2	eed7	f30d	fc25	Avg.
1, 1	③ C/W ①	0.32	0.0	0.0	0.0	0.69	0.35	0.79	0.22	0.0	0.23	0.58	
	③ C/W ②	0.69	0.85	0.69	0.0	0.76	0.0	0.0	0.0	0.90	0.0	0.49	
1, 2	③ C/W ①	0.64	0.69	0.0	0.09	0.85	0.05	0.51	0.0	0.55	0.47	0.74	
	③ C/W ②	0.64	0.68	0.84	0.05	0.86	0.0	0.50	0.0	0.93	0.0	0.73	
1, 3	③ C/W ①	0.0	0.62	-0.03	0.0	0.39	0.56	0.53	0.0	0.05	0.53	0.51	
	③ C/W ②	0.86	0.0	-0.03	0.0	0.38	0.0	0.44	0.0	0.28	0.0	0.54	
1, 4	③ C/W ①	0.38	0.30	0.91	0.08	0.63	0.03	0.80	0.0	0.66	0.80	0.0	
	③ C/W ②	0.99	0.69	0.90	0.0	0.74	0.0	0.57	0.69	0.80	0.0	0.0	
1, 5	③ C/W ①	0.33	0.33	0.0	0.04	0.39	0.32	0.93	0.0	-0.03	0.93	0.87	
	③ C/W ②	0.32	0.83	0.0	0.0	0.37	0.0	0.96	0.0	-0.31	0.0	0.89	
1, 6	③ C/W ①	0.0	0.0	0.75	0.07	0.0	0.34	0.67	0.0	0.42	0.99	0.84	
	③ C/W ②	-0.14	0.96	0.71	0.0	0.15	0.0	-0.07	0.41	0.52	0.0	0.84	
1, 7	③ C/W ①	0.30	0.0	0.56	0.04	0.0	0.525	0.99	0.0	0.29	0.90	0.49	
	③ C/W ②	0.78	0.15	0.69	0.0	0.10	0.0	0.43	0.0	0.42	0.0	0.77	
2, 1	③ C/W ①	0.30	0.56	0.36	0.10	0.51	0.0	0.88	0.0	0.41	0.89	0.85	
	③ C/W ②	0.45	0.77	0.37	0.0	0.57	-0.02	0.0	0.63	0.51	0.0	0.81	
	③ C/W ④	0.85	0.76	0.56	0.10	0.48	0.11	0.90	0.78	0.74	0.86	0.46	
	④ C/W ①	0.39	0.43	0.48	0.43	0.74	0.0	0.98	0.0	0.58	0.82	0.58	
	④ C/W ②	0.53	0.61	0.45	0.0	0.70	0.18	0.0	0.71	0.57	0.0	0.55	
2, 2	③ C/W ①	0.82	0.21	0.91	0.05	0.47	0.03	0.70	0.0	0.29	0.74	0.36	
	③ C/W ②	0.84	0.75	0.93	0.0	0.90	0.0	0.0	0.87	0.59	0.0	0.81	
	③ C/W ④	-0.02	0.82	0.62	0.09	0.84	0.09	0.70	0.83	0.56	0.85	0.46	
	④ C/W ①	-0.02	0.11	0.66	0.38	0.47	0.77	1.0	0.0	0.43	0.70	0.45	
	④ C/W ②	-0.02	0.83	0.63	0.0	0.92	0.0	0.0	0.96	0.71	0.0	0.46	
2, 3	③ C/W ①	0.70	0.44	0.0	0.0	0.62	0.0	0.90	0.0	0.28	0.99	0.68	
	③ C/W ②	0.66	0.44	0.98	0.0	0.72	0.0	0.0	0.47	0.36	0.0	0.82	
	③ C/W ④	0.68	0.54	0.62	0.91	0.49	-0.18	0.90	0.86	0.39	0.98	0.58	
	④ C/W ①	0.91	0.53	0.0	0.0	0.77	0.0	1.0	0.0	0.88	0.96	0.41	
	④ C/W ②	0.74	0.53	0.82	0.0	0.74	0.0	0.0	0.60	0.79	0.0	0.65	
2, 4	③ C/W ①	0.45	0.0	0.83	0.0	-0.02	0.05	0.64	0.0	0.23	0.67	0.78	
	③ C/W ②	0.14	0.85	0.83	0.0	-0.02	0.0	0.0	0.0	0.26	0.0	0.87	
	③ C/W ④	0.17	0.86	0.84	0.90	-0.02	0.0	0.64	0.84	0.38	0.86	0.92	
	④ C/W ①	0.71	0.0	0.86	0.0	0.80	0.47	1.0	0.0	0.68	0.66	0.76	
	④ C/W ②	0.82	0.51	0.86	0.0	0.80	0.0	0.0	0.0	0.50	0.0	0.85	
2, 5	③ C/W ①	0.59	0.0	0.0	0.0	0.28	0.09	0.60	0.0	0.0	0.73	0.95	
	③ C/W ②	0.0	0.41	0.77	0.0	0.28	0.01	0.0	0.54	0.54	0.0	0.92	
	③ C/W ④	0.48	0.47	0.76	0.16	0.26	0.09	0.59	0.82	0.40	0.43	0.47	
	④ C/W ①	0.5	0.0	0.0	0.0	0.82	0.94	0.99	0.0	0.0	0.38	0.46	
	④ C/W ②	0.0	0.34	0.89	0.0	0.83	0.60	0.0	0.49	0.70	0.0	0.44	
2, 6	③ C/W ①	0.48	0.0	0.90	0.0	0.39	0.0	0.73	0.0	0.0	0.20	0.86	
	③ C/W ②	0.0	0.85	0.96	0.0	0.39	0.02	0.0	0.55	0.83	0.0	0.87	
	③ C/W ④	0.47	0.86	0.92	0.77	0.30	0.0	0.72	0.83	0.86	0.74	0.86	
	④ C/W ①	0.98	0.0	0.95	0.0	0.47	0.0	0.99	0.0	0.0	0.46	0.83	
	④ C/W ②	0.0	0.88	0.89	0.0	0.62	0.69	0.0	0.43	0.95	0.0	0.82	
2, 7	③ C/W ①	0.0	0.0	0.0	0.0	0.30	0.0	0.73	0.40	0.43	0.91	0.86	
	③ C/W ②	0.0	0.41	0.76	0.0	0.30	0.0	0.0	0.0	0.14	0.0	0.86	
	③ C/W ④	0.96	0.47	0.72	0.0	0.24	-0.01	0.42	0.79	0.33	0.92	0.84	
	④ C/W ①	0.0	0.0	0.0	0.0	0.79	0.0	0.67	0.46	0.71	0.94	0.78	
	④ C/W ②	0.0	0.80	0.93	0.0	0.78	0.0	0.0	0.0	0.46	0.0	0.78	
Avg. Week 2	③ C/W ①	**0.48**	**0.17**	**0.43**	**0.02**	**0.36**	**0.02**	**0.74**	**0.0**	**0.23**	**0.73**	**0.76**	**0.39**
	③ C/W ②	**0.30**	**0.64**	**0.80**	**0.0**	**0.45**	**0.0**	**0.0**	**0.44**	**0.46**	**0.0**	**0.85**	**0.36**
	③ C/W ④	**0.51**	**0.68**	**0.72**	**0.12**	**0.37**	**0.01**	**0.70**	**0.82**	**0.33**	**0.81**	**0.66**	**0.52**
	④ C/W ①	**0.50**	**0.15**	**0.42**	**0.42**	**0.69**	**0.31**	**0.94**	**0.07**	**0.47**	**0.70**	**0.61**	**0.48**
	④ C/W ②	**0.30**	**0.64**	**0.78**	**0.0**	**0.78**	**0.21**	**0.0**	**0.46**	**0.67**	**0.0**	**0.65**	**0.41**

1.0, where -1.0 means that the two annotators differ at most and 1.0 means complete similarity. 0.0 denotes that the target method was not used on that specific day.

Comparing the ③ *pure self-recall* method with the ① *in situ button* and ② *in situ app* method we can see that the final results for weeks 1 and 2 are proximate to one another. ③ *Pure self-recall* compared with the ④ *time-series recall* results in the highest similarity of 0.52. The comparison between the ④ *time-series recall* and the ① *in situ button* as well as the ② *in situ app* assisted annotations result in higher similarity than the prior comparison of ③ *pure self-recall* vs. both methods ① and ②. This means that subjects rather agree to the timestamps of the *in situ* methods than to a self-written activity diary as soon as they can visually inspect the accelerometer data.

Visual Time-Series Analysis. Figure 3.10 shows exemplary the time-series of the sixth day of every participant's second week. The four bars that are visible above the accelerometer data are the labels set by the participants for every layer. The order is from bottom to top: ① *in situ button*, ② *in situ app*. ③ *pure self-recall*, and ④ *time-series recall*. Examples of labels that differ with regard to the applied labeling method are marked with red boxes. The x-axis of every subplot represents roughly 8-9 hours of data. Most of the day was not labeled and is therefore categorized as *void*. However, such long periods often contain shorter periods of other activities, like *walking*. This makes it difficult to define a distinguishable *void*-class, which results in

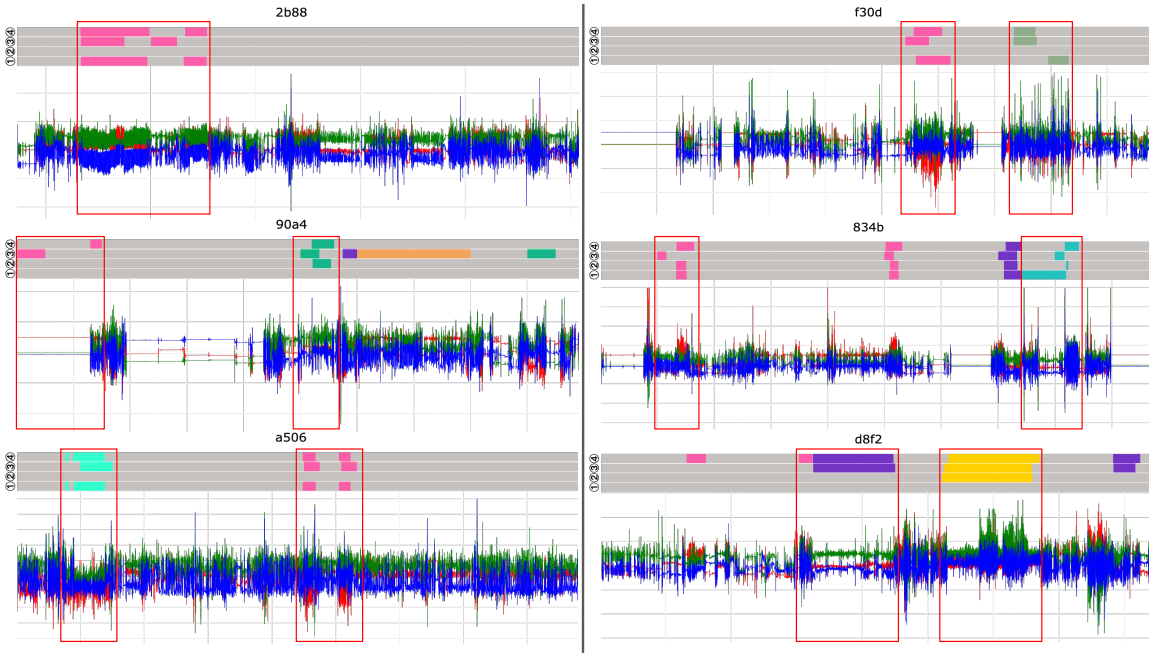


Figure 3.10 Visualization of participants' accelerometer data on the sixth day in the second week of the study, together with annotations set by them. The four layers in the upper part of every participant's daily data represent the four annotation methods. The order is from bottom to top: ① *in situ button*, ② *in situ app*, ③ *pure self-recall* and ④ *time-series recall*.

false positive classifications of non-void samples. Figure 3.10 visually shows that each participant labels his or her data very subjectively. The long green-labeled periods of participant *74e4* represent the class *desk_work*. The only other participant that used this label is *90a4*. Since each of the study participants works in an office environment and thus conclusively works at a desk, we can assume that the same class is classified as *void* for all other study participants. This intra-class and inter-participant discrepancy becomes a problem whenever a model is trained that is supposed to generalize across individuals. To reduce these side effects and focus on the experiment itself, we decided to evaluate personalized models that take weekly data from participants into account.

The *in situ button* annotation is empty for 5 participants: *eed7*, *36fd*, *74e4*, *90a4* and *d8f2*. Labels are only partially set or missing entirely for this annotation method and we therefore assume that participants tend to forget to press the button on the smartwatch. Both tables, 3.1 and 3.2, support this assumption, as this labeling method shows a high percentage of missing annotations as well as a low Cohen κ score of 0.36% (week 1) and 0.39% (week 2). The *pure self-recall* method ③, visible on the 2nd upper layer, is often misaligned compared to the in situ methods as well as the *time-series recall* method ④. Participants tend to round up or down the start- and stop-time in steps of 5 or 10 minutes. For example, the annotations in Figure 3.10 given by the subjects *2b88*, *834b* or *f30d*, show such incorrectly annotated data. The pink color represents the class *walking*. With a closer look at the corresponding time-series data, one can see that the *in situ button* annotation (bottom layer) and *time-series recall* annotation (top layer) belongs to the typical periodic pattern of walking than the period labeled by *pure self-recall*. A consistent reliable performance in all labeling methods can only be observed at the participants *4531* and *fc25*. Other participants like *eed7*, *36fd*, *74e4* or *a506* are very precise in their annotations across methods, but are missing at least one layer of labels. The complete collection of visualizations is available in our dataset repository[4]

Effects on Classification. The results of our deep learning evaluation[5] suggest that the annotation method chosen can have a crucial impact on the classification ability of a trained deep learning model. Depending on the chosen methodology, the average F1-Score results differ by up to 8%, as depicted in Figure 3.11. In the first week, the in situ methodologies, button ① and app ②, generally perform better than the *pure self-recall* diary ③. Study participants mostly correctly estimated the duration of an activity, but tended to round up or down the start and end times. The in situ methods are up to 8% better than the *pure self-recall*, although the amount of annotated data available, due to missing annotations, is significantly lower than for other methods. Although, we work with a dataset recorded in-the-wild, the deep learning results generally show a high F1-Score. This is untypical for such datasets but can be explained by the fact that the majority of the daily data are assigned

[4] `https://doi.org/10.5281/zenodo.7654684`

[5] Detailed results for every participant included in our deep learning evaluation can be accessed online on the Weights & Biases platform: `https://tinyurl.com/4vxvfaed`.

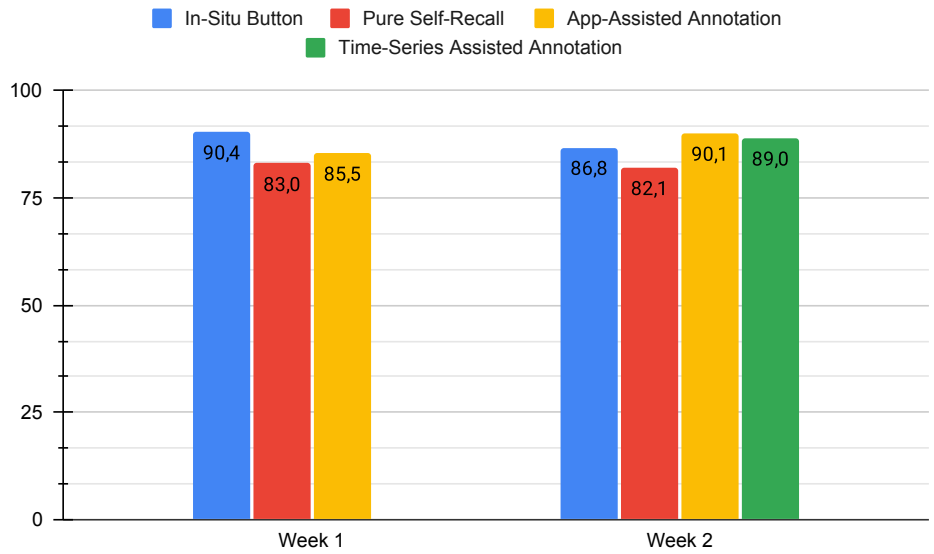


Figure 3.11 The overall mean F1-Scores for the Leave-One-Day-Out Cross Validation across all participants. In the first week, the participants used methods ① - ③. In the second week, we introduced method ④.

to the *void* class. This leaves proportionally only a few samples that are crucial for determining the classification performance.

Even though the number of available annotations that have been labeled by the study participants using the *time-series recall* method ④ is significantly higher with 92.33%, the average F1-Score is 1.1% lower (89.00%) than the results reached with the App Assisted method (90.1%). To understand this result it is crucial to look at Table 3.3 in detail and take meta-information about the participants into account. The participants mostly used their diary as a mnemonic aid for the graphical annotation method and tried to identify the corresponding periods in the acceleration data. The results of subjects *2b88*, *a506* and *eed7* show that the performance of the classifier could be increased with graphical assistance. However, the F1-Score of *2b88* is 0.01% below the F1-Score of the *in situ app* assisted annotation method ②. These subjects have in common that they are already trained in interpreting acceleration data due to their prior knowledge and thus assign samples to specific classes more precisely. Subjects *fc25*, *4531*, and *834b*, on the other hand, do not have prior knowledge. Apart from subject *834b*, the deep learning results show that presenting a visualization to an untrained participant rather harmed than helped the classifier. If one looks at the visualizations of day 1 & 6, week 2 of *fc25*, see Figure 3.10 and 3.12, the labels set by the subject with the help of the graphical interface, it is comprehensible that this study participant tended to be rather confused by the graphical representation and therefore labeled the data incorrectly.

Table 3.3 In detail representation of the final F1-Scores for every annotation methodology and a week per study participant. The average F1-Scores are graphically visualized in Figure 3.11.

Week 1							
Subject	**2b88**	**a506**	**eed7**	**fc25**	**4531**	**834b**	**Average**
① ***in situ button***	0,91	0,92	0,89	0,89	0,91	0,91	90,4
② ***in situ app***	0,92	0,60	0,91	0,84	0,93	0,92	85,5
③ ***pure self-recall***	0,78	0,76	0,83	0,86	0,86	0,89	83,0
Week 2							
① ***in situ button***	0,88	0,92	0,90	0,92	0,86	0,72	86,8
② ***in situ app***	0,91	na	0,92	0,94	0,85	0,88	90,1
③ ***pure self-recall***	0,81	0,86	0,82	0,94	0,83	0,67	82,1
④ ***time-series recall***	0,90	0,91	0,95	0,86	0,86	0,86	89,0

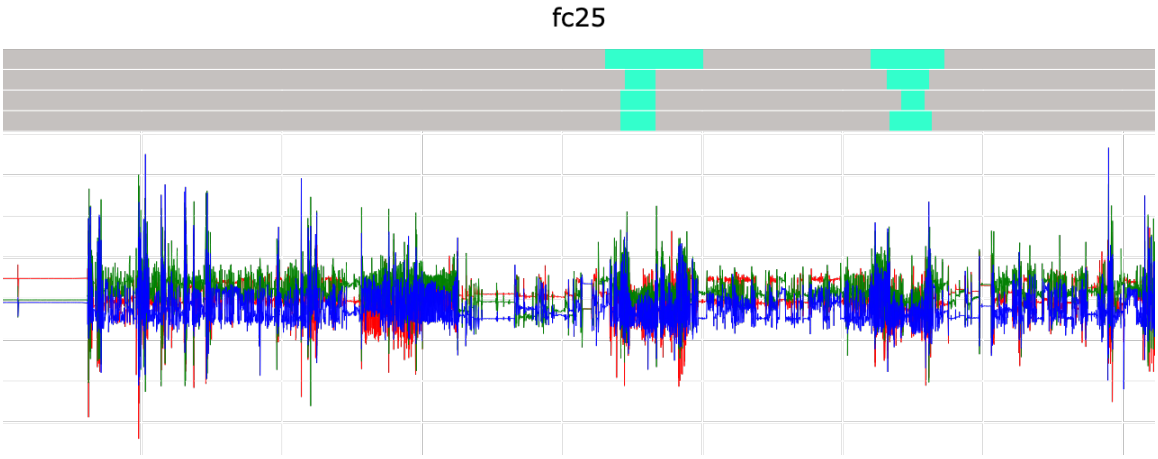


Figure 3.12 Visualization of the 1st day in week 2 of subject *fc25*. Differences can be seen in the upper annotation layer (④ *time-series recall*), exhibiting larger differences regarding the annotated start- and stop times compared to other methods.

3.2.6. Discussion

In our 2-week long-term study, we recorded the acceleration data of 11 participants using a smartwatch and analyzed it visually, statistically, and using deep learning. The findings of the visual and statistical analysis were confirmed by the deep learning result. They show that the underlying annotation procedure is crucial for the quality of the annotations and the success of the deep learning model.

The *in situ button* method ① offers accuracy but brings the risk that the setting of a label is forgotten entirely or incompletely set. However, this method can be combined with additional on-device feedback or a smartphone app, so that greater accuracy and consistency of the annotation can be achieved. This involves a considerable implementation effort, which many scientists avoid because such projects, although of their significant value to the community, attract little attention in the scientific world. The use of existing, but often commercial, software and hardware is all too often accompanied by a loss of privacy. As our research has shown in passing, many users therefore shy away from using such products.

Through our investigation of the consistency of annotations between methodologies,

we were able to show that participants in our study seem to prefer to write an activity diary (*pure self-recall* method ③). This finding corresponds to what Vaizmann *et al.* [213] already points out. However, this method has the disadvantage that it can be imprecise, which is evident in the visualization of the data and annotations. Similarly, the activity diary methodology performed the least reliably among all methodologies, which has been confirmed by the deep learning model. Since the deep learning results using the *in situ app* annotations ③ are almost similar to the results given by the *time-series recall* ④, even though the number of labeled samples is lower, it raises the question if a smaller set of high-quality annotations is more valuable for a classifier than a larger set of annotated data that comes with imprecise labels. This could mean that in future works we can reduce the amount of necessary training samples drastically if a certain annotation quality can be assured. However, this needs to be confirmed by further investigations.

Some participants reported that they found the support provided by the visual representation of the data helpful. The resulting Cohen κ scores strengthen this impression since the F1-Scores are much higher when we compare the *time-series recall* with both in situ methods vs. the *pure self-recall*. This indicates that as soon as the participants received a visual inspection tool, they tended to annotate data at similar time periods as through the in situ methods since they can easily identify periods of activity that roughly correspond to the execution time they remember. Our participants reported similar preferences, which led us to the conclusion that a digital diary that includes data visualization could combine the benefits of both annotation methods.

However, the study also showed that participants can find it difficult to interpret the acceleration data correctly and thus set inaccurate annotations. As our trained models show, this also has a strong influence on the classification result. If such a tool is to be made available to study participants, it must be ensured that they have the necessary knowledge and tools to be able to interpret these data. Thus, to ensure the success of future long-term and real-world activity recognition projects, prior training of the study participants regarding data interpretation is of crucial importance if a data visualization is supposed to be used.

Apart from trying to solve annotation difficulties during the annotation phase itself, we can also partially counter wrong or noisy classified data by using machine learning techniques like Bootstrapping, see Miu *et al.* [143] or using a loss function that specifically tries to counteract this problem, such as [152, 130]. By using Bootstrapping, the machine or deep learning classifier is initially trained by a small subset of high-confident labels and further improved by using additional data. However, this technique comes with the trade-off that whenever wrong-labeled data is introduced as training data, the error will get propagated into the model. An effect that sooner or later occurs as long as the annotation methodologies themselves are not further researched. Other machine learning techniques that can work with noisy labels, see [191], are already successfully tested for Computer Vision problems and can, in theory, be adopted for Human Activity Recognition. However, earlier research has shown that not every

technique that is applicable in other fields is also applicable to sensor-based data [95].

Discussing different Annotation Biases. We need to take into account that several biases could have been introduced due to the chosen annotation guidelines and tools. For example, the usage of in situ annotation methods during the day can have a positive effect on the self-recall capabilities of a participant at the end of the day. However, the comparison of consistencies across methods does not confirm that this effect indeed occurred. Every study participant showed an almost complete overall profile of self-recall annotations, even though the person has not used or has incomplete in-situ annotations, see Figure 3.9. However, deeper investigations are needed to be able to understand such effects better.

Yordanova *et al.* [234] lists the following 3 biases for sensor-based human activity data: Self-Recall bias [214], Behavior bias [69] and the Self-Annotation bias [234]. We showed that indeed a time-deviation bias (which can be seen as a self-recall bias) has been introduced to annotations created with the *pure self-recall* method ③, and that such a bias affects the classifier negatively. However, visualizing the sensor data can counter this effect because it was easier for participants to detect active phases in hindsight.

A behavior bias can be neglected, because the participants were not monitored by a person or video camera during the day and the minimalistic setup of one wrist-worn smartwatch does not influence one's behavior since the wearing comfort of such a device is generally perceived as positive [161]. A self-annotation bias, a bias that occurs if the annotator labels their data in an isolated environment and cannot refer to an expert to verify an annotation, did occur as well. With the deep learning analysis, we were able to show that the classifier was less negatively impacted by this bias than by time-deviation bias.

3.2.7. Conclusions

We argue that the annotation methodologies for benchmark datasets in Human Activity Recognition do not yet capture the attention it should. Data annotation is a laborious and time-consuming task that often cannot be performed accurately and conscientiously without the right tools. However, there is a very limited number of tools that can be used for this purpose and often they do not pass the prototype status.

Only a few scientific publications, such as [172], focus on annotations and their quality. However, the use of properly annotated data drastically affects the final capacities of the trained machine or deep learning model. Therefore, we consider our study to be important for the HAR community, as it analyzes this topic in greater depth and thus provides important insights that go beyond the current state of science. Table 3.4 summarizes the advantages and disadvantages of every method. To guarantee high-quality annotations in future studies, especially in an uncontrolled real-world environment where a video recording is not available for the subsequent acquisition of

Table 3.4 Comparison of advantages and disadvantages of all annotation methods used in this study.

Methodology	Advantages	Disadvantages
① *in situ button*	- Easy to implement and use - Can be improved with feedback mechanisms	- Participants tend to forget pressing a button - Many incomplete annotations that become unusable for the classifier
② *in situ app*	- Tracking apps are already widely used and accepted, therefore low acceptance threshold - Can be improved with feedback mechanisms or additional smartphone functionalities - List of possible annotations can be expanded with minimum effort - Participants tend to set very precise annotations.	- Data and privacy concerns if a commercial app is used. - Participants often forgot to set an annotation, especially when they were unfamiliar with tracking apps. - Implementation workload may be very high
③ *pure self-recall*	- Easy to use even without technical knowledge (a handwritten diary) - Most accepted method in our experiment - Annotations are very consistent	- Can be very imprecise - Only suitable for coarse activity labels and activities that were performed for long periods of time, like *walking* or *running*
④ *time-series* recall	- Visualization of data helps participants to set annotations more accurate than using the pure self-recall method ③	- Available tools are often in the state of a prototype and need additional developments and adjustments and are therefore not impromptu usable. - Participants need to be trained to be able to interpret sensor data.

ground truth, further research in this field is necessary. The methodologies used for annotating activity data need to be challenged and further developed.

The combination of a (handwritten) diary with a correction aided by a data visualization in hindsight shows the best results in terms of consistency and missing annotations and provides accurate start and end times. However, this combination results in additional work for the study participants and therefore, remains a trade-off between additional workload and annotation quality.

Lesson Learned. During this study, we gained insights about the effects of different annotation methods on the reliability and consistency of annotations and finally on the classifier itself, but also about training deep learning models on data recorded in-the-wild. In this chapter, we would like to share these insights to help other researchers perform their experiments more successfully. With regards to Table 3.4, we are able to narrow down specific study setups that either benefit more from self-recall or in situ annotation methods.

(a) Due to the good acceptance and the low workload for study participants we can recommend a self-recall method for studies where label precision is not the highest priority and rough estimations of activities are sufficient.

(b) According to our study, we can increase the label precision of the self-recall method with additional software that visualizes the raw data, e.g. [157]. We recommend considering the implementation of such a module and providing this software to participants together with an introduction on how to interpret sensor data. According to [213] the self-recall method can also be effectively improved by introducing server guesses of activities or visually organizing the day chronologically.

(c) In situ annotations result generally in more precise labels. However, the label process is more labor intensive than a self-recall method, since it can take a lot of time and often includes many steps to set the label. We argue, that smaller

studies with participants who agree with performing such laborious work can benefit from this method. Such a system needs to be implemented carefully and with a holistic concept in order to not be seen as a burden by the participants [233].

(d) Annotating data with commercial apps, like Strava, is negligible due to data and privacy concerns.

(e) In situ annotation can have the same benefits as an app solution. However, only if researchers have access to the programming interface of the recording device and can implement additional features that help participants not forget to set a label.

As part of our annotation guidelines, we allowed our study participants to name their activities as they wished. Therefore, we were forced to simplify certain activities. To be able to create a real-world dataset that contains complex classes or even classes that consist of several subclasses, more elaborated annotation methods and tools must be developed. We believe that with the currently available resources, the hurdle lies very high for such datasets to be annotated accurately.

Our study includes people who cycle to work in their daily work routine and others who commute by public transport or work in a home office environment. Thus, each study participant has his or her set of daily repetitive activities. Due to the nature of our dataset as one recorded in a real-world and long-term scenario, the number of labeled samples is rather small, and given labels vary participant-dependent. This mix of factors creates a bias in the dataset and we concluded that a cross-participant train-/test-strategy is not appropriate for our study design and would not give meaningful insights, since every study participant has their own set of unique activities which are too different and hardly generalizable. Therefore, for certain studies, the commonly known and accepted Leave-One-Subject-Out Cross-Validation is not suitable.

Section 3.3

Multi-Sensor Synchronization

[91] Hoelzemann, Alexander, et al.
Using an in-ear wearable to annotate activity data across multiple inertial sensors
Feb. 2020, EarComp'19: Proceedings of the 1st International Workshop on Earable Computing
https://doi.org/10.1145/3345615.3361136

Portions of the original publication have been removed or edited for inclusion in this thesis. However, no changes were made that altered the results or conclusions presented in the original work.

Contributions:

- Kristof Van Laerhoven and I designed the study.
- I implemented and executed the experiments.
- Henry Odoemelem developed the eSense recording system.

Wearable activity recognition research needs benchmark data, which rely heavily on synchronizing and annotating the inertial sensor data, to validate the activity classifiers. Such validation studies become challenging when recording outside the lab, over longer periods. This section presents a method that uses an inconspicuous, ear-worn device that allows the wearer to annotate his or her activities as the recording takes place. Since the ear-worn device has integrated inertial sensors, we use cross-correlation overall wearable inertial signals to propagate the annotation's overall sensor streams. In a feasibility study with 7 participants performing 6 different physical activities, we show that our algorithm can synchronize signals between sensors worn on the body using cross-correlation, typically within a second. A comfort rating scale study has shown that attachment is critical. Button presses can thus define markers in synchronized activity data, resulting in a fast, comfortable, and reliable annotation method.

3.3.1. Introduction

As wearable sensors have been shrinking and getting less power-hungry, their operation time and places where they can be worn have inadvertently increased accordingly. Nowadays, multiple such sensors can be worn as patches or miniature straps anywhere on the limbs, torso, or even on the head. When doing experiments with such sensor data, however, the annotation of the data has remained a burden, taking a substantial amount of effort. Few methods exist that allow the sensor data to be annotated directly, even fewer methods allow these annotations to be made for any amount of wearable sensor data from the user's body. We argue that an in-ear device that is

equipped with inertial sensors and a button would be an excellent candidate for user annotation of activity data, as shown in Figure 3.13. It would allow the users to

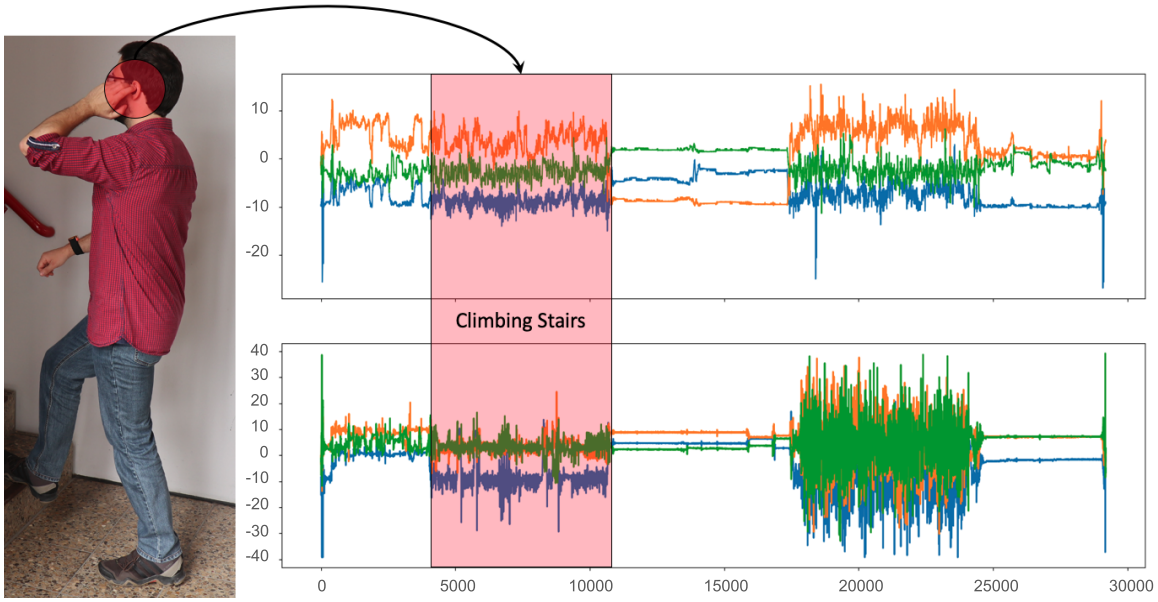


Figure 3.13 We present an inconspicuous annotation method, in which users can annotate their activity data in situ with an in-ear wearable (left), to mark and synchronize inertial data from the ear with all other inertial sensors (right).

annotate their data without much effort in a socially comfortable way, which also enables in-the-wild experiments as study volunteers annotate activities in their daily lives. A critical step in our method is the synchronization of sensor data between all wearable sensors: We assume that all sensors contain inertial sensors that show sufficient correlation during everyday activities. The synchronization of different sensor signals plays a decisive role in activity recognition. In most cases, a synchronization gesture is executed at the beginning and end of the measurements to synchronize the two or more time series. This method has the decisive disadvantage that it is time-consuming and error-prone. With this section, we would like to introduce another approach, that helps synchronize an arbitrary number of sensor signals. These signals only have the basic preconditions that they must be recorded at the same time and that sufficient sedentary phases, greater than 1 minute, are included. The presented algorithm works with very few calculation steps, these are the calculation of the vector length, the standard deviation, and a binary filter that is used to decide if the acceleration signal represents a sedentary or non-sedentary activity. The given results show a median time mismatch of 1.10 seconds and can be used to synchronize related, but independently captured, sensor signals with a shared time base. In order for the algorithm to work reliably with the raw data, it must first be prepared and preprocessed. Our presented algorithm is fast and easy to implement. This allows researchers to take up this idea and incorporate it into their projects [30].

3.3.2. System Design

Hardware. The hardware used for collecting labeled data is the eSense-BLE [105] by the Pervasive Systems group at Nokia Bell Labs, Cambridge. It is built with a custom-designed 15 x 15 x 3 mm PCB and composed of a Qualcomm CSR8670, a dual-mode Bluetooth audio system-on-chip (SoC) with a microphone per earbud; an InvenSense MPU6500 six-axis inertial measurement unit (IMU) including a three-axis accelerometer, a three-axis gyroscope, and a two-state button; a circular LED; associated power regulation; and battery-charging circuitry. It is powered by an ultra-thin 40-mAh LiPo battery but lacks internal storage or a real-time clock. Each earbud weighs 20g and is 18 x 20 x 20 mm. The left earbud is the one containing the IMU sensor accessible through the BLE and will be used in the remainder of this section. The Platypus prototype is a wrist-worn activity sensing platform [89] that is equipped with a number of sensors, including a full MPU9250 IMU, environmental sensors, and several processing units included in an Edison System-on-Chip module that runs an embedded Linux distribution as the operating system. We have used this prototype as it can record the IMU data at a relatively high sampling speed of 300 Hz and present the recordings via a Secure Shell (SSH) over the built-in WiFI transceiver.

Mobile Application. To be able to get labeled data for cross-correlation, we developed an Android App for data collection on Android Studio by adapting only the needed aspects of the Android library provided by the developers. The Android ScanFilter was used to restrict the scan result to our desired eSense device, using the LOW LATENCY scan mode. The notification was enabled by writing to the descriptor for the push button status and accelerometer data from the onboard IMU, which we set to a 50Hz sampling rate. The accelerometer works with the default configuration of +/-4g (sensitivity of 8192 LSB/g) . Using the Android onCharacteristicChanged, accelerometer data about the three-axis is received and checked for correctness using the CheckSum, then stored in the internal storage of our mobile phone as a CSV file in units of g and multiplied by 10 to increase the amplitude. We also saved a time-stamp in microseconds that have elapsed since January 1, 1970, at 00:00:00 GMT. Additionally, on every button push, the current data from the accelerometer is labeled with an ASCII character and stored, as well as displayed on the TextView. We had a challenge of receiving the same data on different timestamps, but this was resolved by keeping the processing time in the onCharacteristicChanged method as low as possible, another problem that we encountered, was that each button push notification caused some accelerometer package index to be skipped on subsequent readings, restarting the IMU sampling on each button push solved this problem. Finding and establishing a connection with the eSense (BLE and classic Bluetooth) is a challenge and requires several trials.

3.3.3. Methodology

Besides our study about the reliability of our proposed algorithm, we also asked the participants to fill out a short questionnaire regarding the wearing comfort of the eSense earbud by using the Comfort Rating Scale (CRS) as proposed in [109].

Dataset. A data set of activity data of seven participants has been recorded using the eSense and the platypus. The data recorded by the eSense is sampled at 50 Hz, and the data recorded with the platypus is sampled at 300 Hz. The participants are between 20 and 40 years old. We were able to recruit five men and two women for the study. The platypus data set consists of a total amount of around 2.255.764 samples or 2.08 hours of data. For the eSense we recorded 375.967 samples, which also resulted in 2.08 hours. The data set contains acceleration data from both sensors for mixed activities: (1) reading or desk work, (2) walking, (3) climbing stairs, (4) sitting, (5) dribbling a basketball, and (6) pause or rest phase.

eSense Wearing Comfort. In addition to the evaluation of the reliability of our presented algorithm, it was also very important for us to evaluate how comfortable the provided prototype was perceived by the participants of the study. Table 3.5 shows the categories and their description. The Emotion, Harm, and Anxiety categories

Table 3.5 Comfort Rating Scale (CRS) categories, as proposed in [109]. The CRS includes 6 categories: Emotion, Attachment, Harm, Perceived change, Movement, and Anxiety.

Title	Emotion	Attachment	Harm
Description	The device is causing me some harm. The device is painful to wear	I can feel the device on my body. I can feel the device moving	I am worried about how I look when I wear this device. I feel tense or on edge because I am wearing the device.

Title	Perceived Change	Movement	Anxiety
Description	Wearing the device makes me feel physically different. I feel strange wearing the device.	The device affects the way I move. The device inhibits or restricts my movement.	I do not feel secure wearing the device.

are more about personal and psychological sensations when wearing the device, while the remaining three categories focus on the device's body feel. The participants can choose a value between 0 and 10 for every category. 0 means it has a low impact, and 10 is a high impact.

Data Synchronization. For evaluation purposes, a ground truth dataset is required. Therefore, we utilized a synchronization gesture. The synchronization gesture required both hands and the head to exhibit the same vertical jump movement, generating clearly identifiable acceleration signal peaks. The gesture occurred at the start and conclusion of data recording to timestamp the dataset boundaries. To synchronize two independently recorded signals, data preprocessing was necessary. The first preprocessing step cropped the data between the synchronization gesture timestamps. Due to sample losses from wireless transmission errors or incomplete sampling rates, the

initial signal durations differed. Signals also required matching sampling frequencies. Thus, the Platypus data was downsampled to 50 Hz. As eSense provided the only means of sequence labeling, its timestamps defined the ground truth for all sensors undergoing synchronization. Under these conditions, all body sensor signals could be resampled equidistantly. To simulate lacking synchronization gestures, the start and end of the eSense data was shortened 10%. Since the method for synchronizing signals is of central importance, it takes up most of the work presented here. Table 3.6 describes step-wise the developed algorithm and the results after every step. When the algorithm finishes we are able to propagate the label throughout the sensors. Parameters like the window size and window length, but also the threshold of the binary filter, can be adjusted variably. In the first version of the algorithm, a simple ASCII character is written with a button press at the beginning and end of the activity. In the future, we plan to use the microphone and voice-to-speech recognition for setting the label.

3.3.4. Results and Discussion

Comfort Rating Scale. Figure 3.14 shows the result of our CRS study. To sum the result up we can say that in general the device is comfortable to wear, but sometimes

Table 3.6 Step-by-step explanation of the algorithm. The algorithm is divided into 8 steps. First, the dimension of the data is reduced by calculating the vector length, divided into windows and finally, the standard deviation is calculated. The standard deviations are now passed through a binary filter, which writes a 0 for sedentary activity and a 1 for non-sedentary activity. Both signals are then cross-correlated. The position of the highest correlation can then be used to deduce the synchronization point in the initial signal.

Step	**1**	**2**	**3**	**4**
Name	Dimension Reduction	Windowing	Feature Extraction	Binary-Filter
Description	Calculate vector length per sample (dimension reduction from 3D to 1D).	Data is divided into windows. Window length and overlap ratio can be set variable.	Calculation of the standard deviation.	Both standard deviation signals are passed through a binary filter. A threshold is used to decide whether it's a sedentary or a non-sedentary activity. 0 (sedentary) if the current value is smaller than the threshold and 1 (non-sedentary) if higher than the threshold.
	Result: Normalized signals, with reduced dimension.	Result: Windowed data.	Result: Standard deviation per window.	Result: Two signals with the values 0.0 and 1.0 for sedentary sequences and 1 or non-sedentary activities.

Step	**5**	**6**	**7**	**8**
Name	Cross-Correlation	Index Selection	In-Window-Cross-Correlation	Label Propagation
Description	Cross-correlation [3] of both binary filtered signals. The eSense signal is cross correlated with the other signals.	The window with the highest correlation coefficiant marks the best index to synchronize the signal.	Cross correlation for all samples in this window.	Labels from the eSense signal can be copied to the other sensor data.
	Result: Cross-Correlation coefficient	Result: Start window for synchronizing	Result: Exact index for synchronization.	Result: Labeled data.

you can feel it moving in your ear. One participant in the study noted that the earplugs tend to fall out of the ear during heavy movements, like dribbling a basketball, even if adjusted correctly. The average values for the Emotion, Harm, and Anxiety categories show that users are generally not concerned about their appearance. This is certainly due to the fact that earbuds are very inconspicuous and devices like these have long since found their way into our everyday lives. The results in the other categories

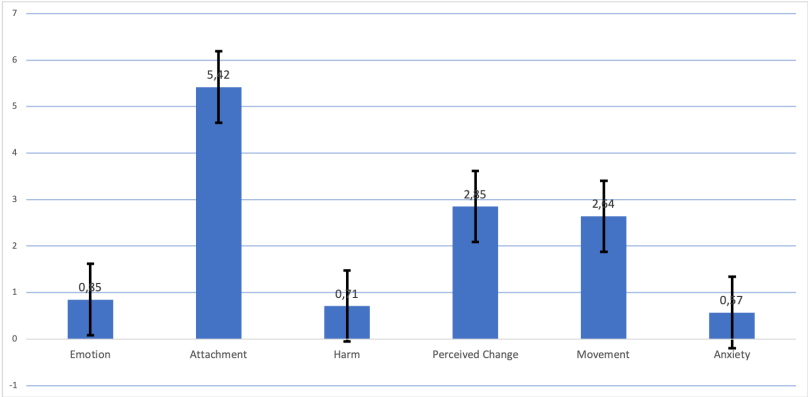


Figure 3.14 CRS result means and standard deviations. Anxiety: 0.57, 0.6; Movement: 2.64, 3.09; Perceived Change: 2.85, 3.17; Harm: 0.71, 0.8; Attachment: 5.42, 2.38; Emotion: 0.85, 1.31.

vary. This shows the standard deviation. The perceived wearing comfort is strongly user-dependent and is probably also related to the individual shape of the inner ear. This is as unique as the fingerprint [58], which is why it is difficult to develop a shape that everyone feels comfortable with.

Data Synchronization Method. In order to investigate the reliability in terms of automatically synchronizing the inertial data streams, we decided to use the time and sample mismatch between the ground truth and the index used as the synchronization point. To evaluate the performance of our algorithm we first calculated the best working parameters for window size and overlap ratio by using a brute force method. This was possible because of the short computational time and, compared to long-term benchmark data, a limited amount of data. The determined parameters from these experiments were found to be: window-size 50, overlap-ratio 85%. With these parameters fixed, we calculated the time mismatch separately for every inertial data recording, as depicted in Table 3.7. The graphical representation as given out by our algorithm is shown in Figures 3.15, 3.16, and 3.17. These figures present two different signals: The top one is the one recorded at the wrist by the Platypus prototype. The bottom plot contains the inertial data from the eSense. The ground truth, as obtained from synchronization gestures before and after the recording (not shown), as a reference is plotted transparently. Overlaying the ground truth the resulting shortened and synchronized signal is depicted, with the vertical red lines marking the beginning and ending of the calculated synchronization point. The black line plot embedded in each bottom plot shows the correlation signal. The inertial signals are synchronized according to the highest cross-correlation. In the first version of the algorithm, the binary filter was not yet part of it, which resulted in problems synchronizing correctly,

Table 3.7 Synchronization error per record in samples and seconds. The records are from 7 participants and 6 activities. Synchronization tends to be within one second for records with clear sequences of different intensities: (1) read or desk work, (2) walk, (3) climbing stairs, (4) sit, (5) dribble basketball, and (6) pause or rest.

Record	Mismatch in Samples	Mismatch in Seconds	Activity
1	15	0.30	1, 3, 5, 6
2	16	0.32	1, 6
3	16	0.32	1, 4, 6
4	20	0.40	1, 6
5	21	0.42	1, 2, 5, 6
6	21	0.42	1, 4, 6
7	23	0.46	1, 3, 5, 6
8	87	1.47	1, 3, 5, 6
9	293	5.86	1, 3, 5, 6
10	386	7.72	1, 4
11	418	8.36	5
12	874	17.48	1, 4
13	1195	23.90	1, 2, 3, 5
14	1742	34.84	1, 4

if no pause phases have been part of the record, for example, record 11 in Table 3.7. The binary filter sets a very hard boundary between sedentary and non-sedentary activities, decided by a threshold, wherefore we needed to have a closer look at the calculated standard deviation signal. Here we saw that the threshold needs to be between 0.500 mg and 0.515 mg. After setting the boundaries we evaluated that the best working threshold is at 0.508 mg. The mismatch (median) of the algorithm was 61 samples or 1.22 seconds. Due to the usage of the binary filter, we were able to improve our results to 55 samples or 1.10 seconds of mismatch. The records that could be rather poorly synchronized with our algorithm are records that mostly consist of sedentary activities as e.g. sitting, reading, or desk work, as depicted in Figure 3.16 or records with heavy movements, but without pause phases, Fig. 3.15. Very well

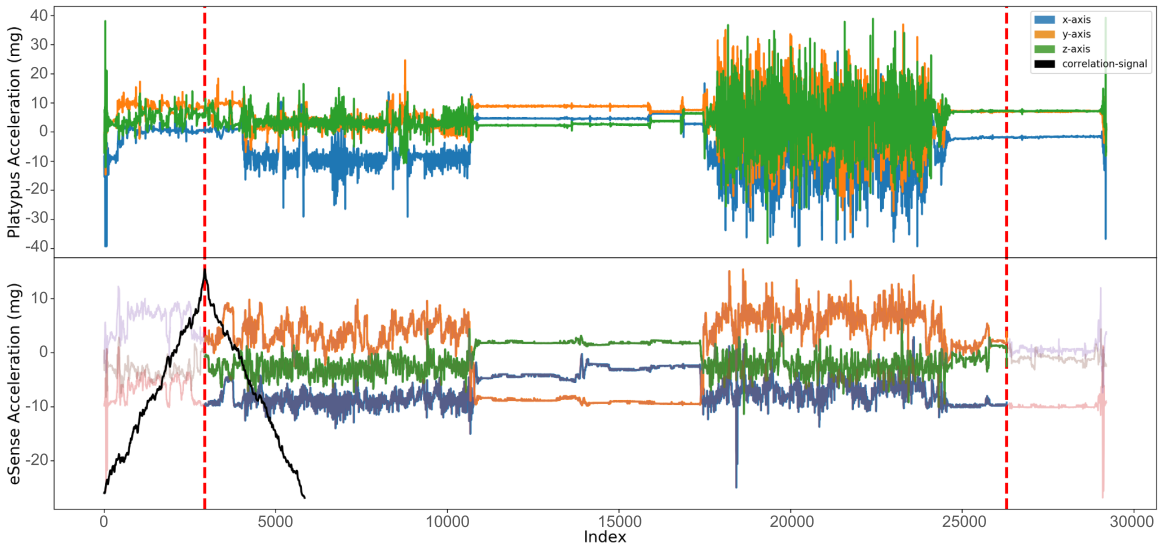


Figure 3.15 Best-case synchronization with a mismatch of 0.30 seconds. The figure shows that synchronization works best with sufficient long periods of sedentary activity. The inertial signal of the wrist (top) is compared with the synchronized signals of the head (bottom). The black signal at the bottom left depicts the cross-correlation between the binary-filtered signals.

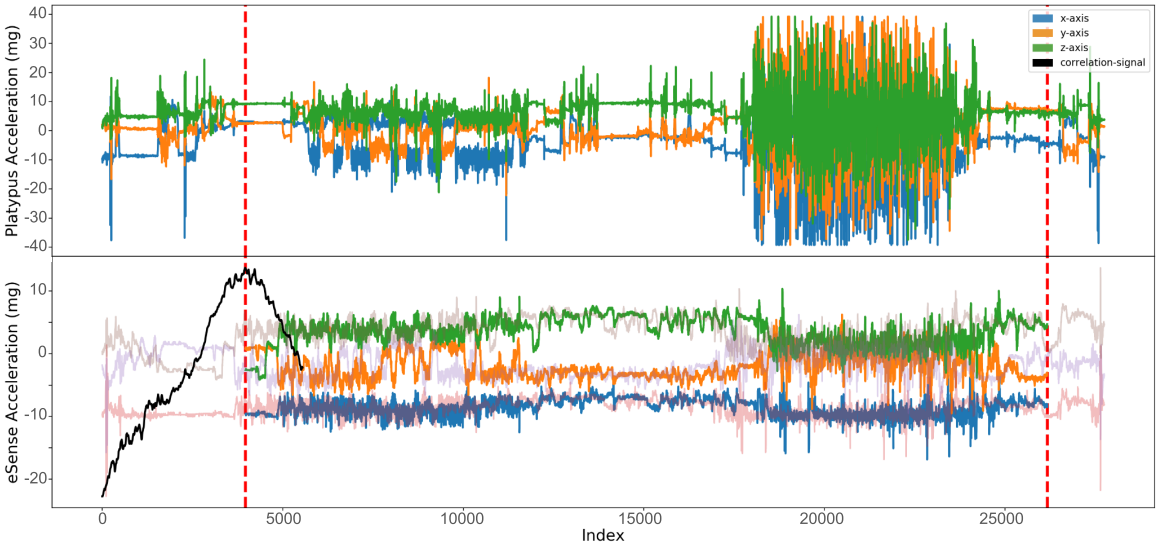


Figure 3.16 Data without sufficient pause phases, with plots defined as in Figure 3.15, using Record 13 in Table 3.7. Our algorithm's synchronization was off by 1195 samples or 23.90 seconds.

devoted, data sets can be synchronized that reflect activities involving a high degree of locomotion as well as sufficient phase of pauses, e.g. Figure 3.15.

(a) **Minimum time mismatch:** 0.30 seconds or 15 samples
(b) **Maximum time mismatch:** 34.84 seconds or 1742 samples
(c) **Median time mismatch:** 1.10 seconds or 55 samples

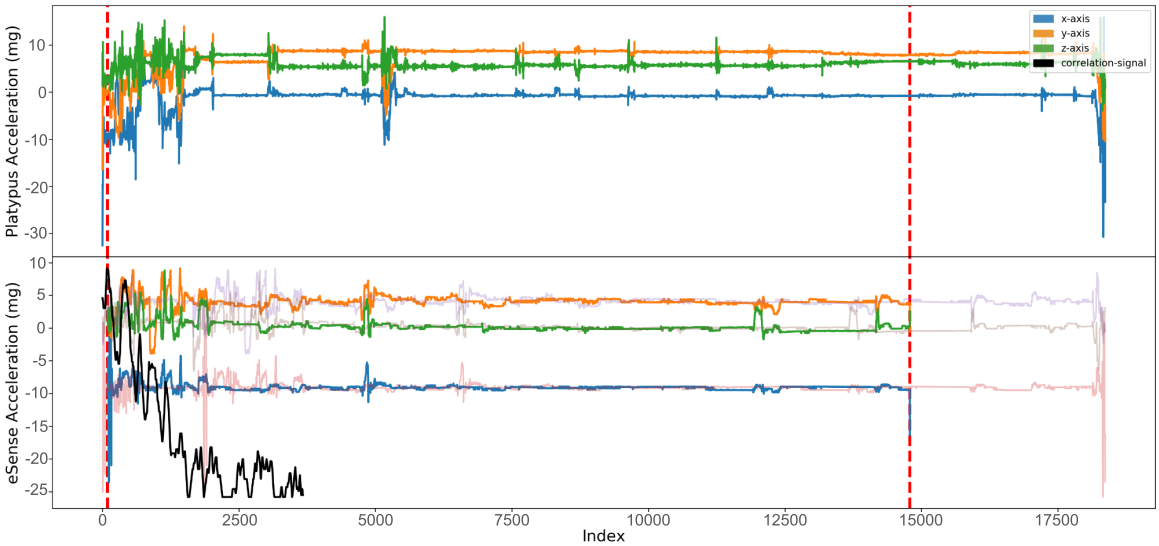


Figure 3.17 Worst-case synchronization with a mismatch of 34.84 seconds, record 14 in Table 3.7. In this data, mostly desk work has been performed. The inertial signal of the wrist (top) is compared with the synchronized signals of the head (bottom). The black signal at the bottom left depicts the cross-correlation between the binary-filtered signals.

3.3.5. Conclusions

We presented in this section a novel annotation method for recording activity recognition benchmark data. Our method relies on users wearing a small earbud-like device in their ear, which is equipped with a button and an inertial measurement unit. The inertial data from the ear-worn sensor are synchronized to all other data via cross-correlation, after which the user presses serve as labels that annotate all sensor streams. In a preliminary study with 7 users, we investigated how well this synchronization works, as well as how comfortable the earbud-like wearable was to our study volunteers. This paper offers a first approach to spread the annotations temporally correct over any number of sensors and to synchronize time series that have been recorded at the same time from different devices. If the data contains sequences that can be uniquely assigned to an activity, with sufficient periods of resting activity, the synchronization was found to be sufficiently reliable. However, the algorithm does not work reliably enough if the head and hand movements during an activity do not basically follow the same direction or if they can completely differ from each other. In addition, care must be taken to ensure that the movements follow a pattern that includes rest periods. The evaluation, as in Table 3.7, has shown that these are essential for reliable synchronization. In terms of wearing comfort, we found that the used eSense prototype is highly promising as an annotation tool for everyday recordings in-the-wild. The fact that it can be worn comfortably, with attachment as the weakest link for some participants, and almost hidden in the ear makes it ideal for recording and annotating data outside our laboratory. As such devices could be operated simultaneously as wireless headsets, the one remaining hurdle for the use of our method in long-term and day-long activity recordings is the eSense's battery.

Section 3.4

The Hawthorne Effect in Sensor-Based Human Activity Recognition

[90] Hoelzemann, Alexander, et al.
A Data-Driven Study on the Hawthorne Effect in Sensor-Based Human Activity Recognition
Oct. 2023, HASCA23: Proceedings of the 11th International Workshop on Human Activity Sensing Corpus and Applications
https://doi.org/10.1145/3594739.3610743

Portions of the original publication have been removed or edited for inclusion in this thesis. However, no changes were made that altered the results or conclusions presented in the original work.

Contributions:

- All authors contributed to designing the study.
- Ericka Andrea Valladares Bastías, Salma El Ouazzani Touhami and Kenza Nassiri and I recorded the dataset.
- Marius Bock and I analyzed the data.
- Kristof Van Laerhoven guided this work and assisted in the methodologies.

Known as the Hawthorne Effect, studies have shown that participants alter their behavior and execution of activities in response to being observed. With researchers from a multitude of human-centered studies knowing of the existence of the said effect, quantitative studies investigating the neutrality and quality of data gathered in monitored versus unmonitored setups, particularly in the context of Human Activity Recognition, remain largely under-explored. With the development of tracking devices providing the possibility of carrying out less invasive observation of participants' conduct, this study provides a data-driven approach to measure the effects of observation on participants' execution of five workout-based activities. Using both classical feature analysis and deep learning-based methods we analyze the accelerometer data of 10 participants, showing that a different degree of observation only marginally influences captured patterns and predictive performance of classification algorithms. Although our findings do not dismiss the existence of the Hawthorne Effect, it does challenge the prevailing notion of the applicability of laboratory compared to in-the-wild recorded data.

3.4.1. Introduction

Body-worn sensor systems bear great potential in analyzing our daily activities with minimal intrusion, yielding various applications from the provision of medical support to supporting complex work processes [45]. With (deep) neural networks representing

the state-of-the-art technology for the automatic analysis of such wearable data, a bottleneck becomes the correct annotation of data for the underlying training. Due to the fact that inertial data is difficult to interpret in hindsight without any additional context, most publicly available datasets remain captured in controlled, video-recorded environments with researchers being in close proximity of study participants. The Hawthorne Effect, originally discovered in 1958 [112], describes the phenomenon that humans alter their behavior and execution of activities in response to being observed. The phenomenon's discovery has led to numerous studies trying to measure the said effect in clinical trials [25, 139, 219, 175, 134, 167], and, more targeted toward physical activities, showed that the effects can be quantified, for instance with gait parameters like step length and cadence of gaits [219]. With researchers from a multitude of human-centered studies being aware of the existence of such an effect, a data-driven study of the phenomenon and its potential effects remain largely under-explored in the community of Human Activity Recognition. Given that the performance and applicability of learning algorithms, such as neural networks, in real-world scenarios heavily depend on the representativeness of the training data, our study aims to investigate whether the prominent observation of participants during data collection introduces biases and results in measurable and altered executions of activities which, in turn, may potentially lead to less effective and less generalized networks. Inspired by the works of Vickers *et al.* [219], our paper provides a data-centric analysis of measuring a possible Hawthorne Effect on a variety of fitness activities through the modality of wrist-worn inertial sensor data. This is done by explicitly letting the participants be observed through cameras and/or the researchers. Contributions of our paper are three-fold:

(a) We designed a HAR experiment where volunteers perform a set of activities under three observation settings: 1) fully-observed (video-recorded + monitoring by researchers), 2) semi-observed (video-recorded + no monitoring), and 3) non-observed (no video recording + no monitoring).

(b) We collected data from 10 participants performing 5 different activities, *jumping, walking, jogging in place, sit-ups, and jumping jacks* over several days.

(c) We perform both feature analysis and investigation of changes in the predictive performance of a deep learning classifier [35] based on the type of observation applied during validation as well as its capabilities to distinguish between each participant's session.

3.4.2. Methodology

Study Protocol: To investigate any potential effects observation of participants can have on collected inertial data, we asked 10 participants (4 females, 6 males) to perform a short workout across multiple days, employing different types of observation (see Figure 3.18). The study was approved by our university's ethics council. Study participation was voluntary, and informed consent was obtained from all participants before the study. The workout plan consisted of a fixed order of 5 different activities, i.e. *jumping*, *walking*, *jogging_in_place*, *sit_ups*, and *jumping_jacks*, each performed for

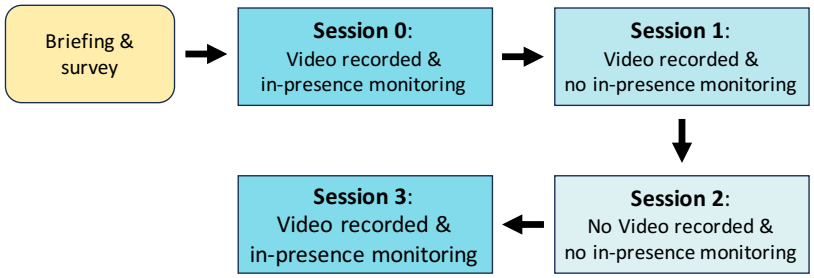


Figure 3.18 Applied study protocol. After having a briefing and filling out a pre-study survey, each participant performed the workout 4 times across 4 different days. The first and last workouts (sessions 0 and 3) were video-recorded and performed under the observation of at least one researcher. The second and third workouts (sessions 1 and 2) were performed without the observation of any researcher at a location chosen by the participant (e.g. at home). The second workout (session 1) was additionally video-recorded.

Table 3.8 Pre-study survey answers provided by each participant. The survey asked participants to provide age, gender, and whether they perform regular private workouts and wear a wearable device (e.g. fitness smartwatch) in their daily lives.

ID	Age	Gender	Pvt. Workouts	Pvt. Wearable
sbj_0	18-25	F	✗	✗
sbj_1	26-35	F	✓	✗
sbj_2	26-35	M	✓	✓
sbj_3	18-25	M	✓	✓
sbj_4	18-25	F	✗	✓
sbj_5	26-35	M	✓	✓
sbj_6	18-25	F	✗	✓
sbj_7	18-25	M	✗	✗
sbj_8	26-35	M	✗	✗
sbj_9	26-35	M	✗	✗

120 seconds with breaks in-between the activities.

Before their first workout session, each participant was briefed about the study protocol in a structured manner and shown sample data collected by the tracking device. Participants were further asked to answer a short survey asking for gender and age group as well as whether they perform regular private workouts and a fitness tracker in their daily lives (see Table 3.8). To avoid any unwanted biases, participants remained unaware throughout the study that the data would be analyzed to assess differences between supervised and unsupervised recording setups. In total, each participant was tasked to perform the workout 4 times using 3 different types of observation. After having been briefed, each participant was equipped with a smartwatch on their wrist of choice and given a demonstration by one of the researchers of the correct execution of each exercise. The smartwatch, a Bangle.js Version 1, was set to record 3D accelerometer data at a constant sampling rate of 12.5 Hz with a sensitivity of ±8g using a custom, open-source firmware[216]. The first workout session (*session 0*) was performed under the observation of one researcher in a location decided by the participant and researchers with a video-recording device taping the execution of the routine. After the completion of *session 0*, participants were walked through the

control of the smartwatch and tasked to perform the workout plan within the next days twice at a location of their choice (e.g. their home) – one-time video-recording (*session 1*) and another time without video-recording their workout (*session 2*). Lastly, participants were invited back to where they originally performed the first session and asked to perform the workout a second time under the observation of a researcher with a video recording in place (*session 3*). In between the 4 sessions participants were asked to wear the smartwatch as much as possible throughout their daily life, keep a brief recount of their daily activities, and note down the start and end times of each of them. To further ensure the workout of interest can properly be identified in the activity streams, each session started and ended with the activity *jumping*. Having identified the workouts in the inertial data recordings, the 3D-acceleration data streams were cropped to only include the workout activities, labeled accordingly, and saved session-wise for each participant into separate files.

Feature Analysis. The feature analysis incorporates a Fast Fourier Transform (FFT), as depicted in Figure 3.19 and a comparison of the total number of repetitions, indicated by the $\sum$-sign, and repetitions per second for a specific activity, indicated by the $\varnothing$-sign, taking into account the subject and session in which the activity was performed. The results are presented in Table 3.9. Both, the FFT [203] and the repetitions per second, calculated with a peak detection algorithm [205], are calculated utilizing functions provided by the SciPy community. The peak detection algorithm specifically processes 1-dimensional time-series data. For our analysis, we computed the magnitude of the accelerometer signal and employed it for the algorithm. Given the periodic nature of the activities under study, each positive peak observed in the signal can be interpreted as indicating one repetition. To validate the accuracy of the peak detection, a visual confirmation was conducted.

Deep Learning Analysis. As proposed by Ordoñez and Roggen in [159], a popular methodology in human activity recognition remains the usage of convolutional and recurrent layers. The former is used to automatically extract discriminative features. Having shown quantitative differences in the feature analysis, the following will investigate the effect said (potential) differences have on the performance and applicability of neural networks. All experiments were conducted using a shallow DeepConvLSTM [35] employing a kernel size of 3, 1024 hidden LSTM units, and inertial data which was split into sliding windows of 1 second with a 50% overlap. We reuse hyperparameters reported in [35], proven to work on a multitude of activity recognition datasets, and only increase the number of epochs (300) while employing a step-wise learning rate schedule, decreasing the learning rate by a factor of 0.9 every 30 epochs. To minimize the risk of performance differences between experiments being the result of statistical variance, reported metrics are averaged across three runs using three different random seeds. Our three types of experiments aim to answer three types of questions: **(1) Cross-session generalization:** How well does a network, trained using fully-observed data, predict data recorded employing different degrees of observation? That is, for each subject, predict each session's activities individually having trained on all other subjects' session 0 data. **(2) Session differentiation:**

Can a network be trained to classify data records into the respective session type they originate from? That is, for each subject, predict each data records session type having trained on all other subjects' data. **(3) Fully-observed overfitting:** Are patterns learned by a network overfitted on a subject's fully-observed data transferable to data employing different degrees of observation? That is, for each subject, predict activities recorded during sessions 1, 2, and 3 using a network overfitted, i.e. reaching close-to-perfect classification scores, on session 0 data. In order to achieve the network overfitting, these experiments involve increasing the number of epochs (1000), learning rate (0.2), and applied learning rate scheduler step size (250).

3.4.3. Results

Overall, we were not able to prove that the Hawthorn Effect is directly verifiable by any of the aforementioned analyzing methods or that data recorded in various recording environments (controlled or semi-controlled) differ significantly.

Feature Analysis. The analysis of the Fast Fourier Transform, Figure 3.19, reveals that fully monitored sessions 0 and 3 generally do not exhibit similar dominant frequencies, which is also the case for semi-monitored and unmonitored sessions 1 and 2. In particular, several activities and subjects align with our previously established hypothesis that the signal from session 3 should converge back to that of session 0. Such behavior is evident for *sbj_6*, *sbj_7* and *sbj_8* during the *jumping_jacks* activity, and for *sbj_2* during the *sit_ups* activity. It is important to acknowledge that this bias may have arisen both due to the researcher's observations and the spatial variations in the workout environment. The fact that this behavior is more evident while executing *jumping_jacks*, might indicate that a Hawthorne Effect is limited to a specific kind

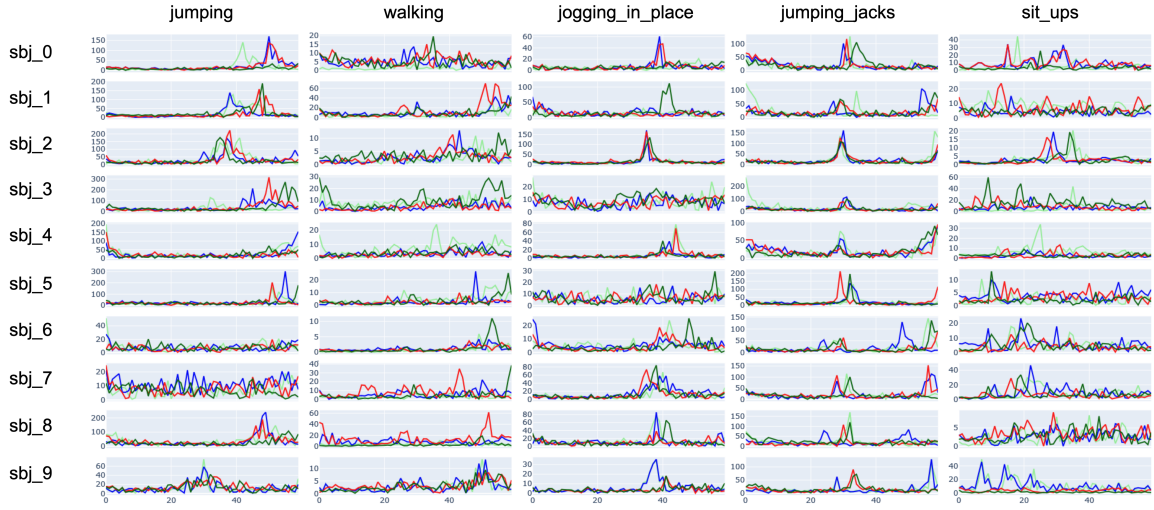


Figure 3.19 Fast Fourier Transform (FFT) calculated on every participant and every activity included in our study. Light-green represents session 0 (monitored and video recorded), blue session 1 (non-monitored, video recorded at home), red session 2 (non-monitored, non-video-recorded), dark-green session 3 (monitored and video recorded)

Table 3.9 This table shows the total number of repetitions ($\sum$) encountered in the activities' signal and the number of repetitions per second ($\varnothing$). Every subject has 4 rows that represent the session. Activities that are marked as ▢ represent cases where the number of repetitions per second was higher during the monitored sessions than during the unmonitored ones, activities colored in ▢ are activities where the number of repetitions per seconds was higher during the unmonitored sessions than during the monitored sessions.

	jumping $\sum$, $\varnothing$	**walking** $\sum$, $\varnothing$	**jogging_in_place** $\sum$, $\varnothing$	**jumping_jacks** $\sum$, $\varnothing$	**sit_ups** $\sum$, $\varnothing$
sbj_0	41, 1.68	39, 1.77	79, 1.92	32, 1.01	20, 0.61
	51, 1.66	41, 1.28	43, 1.44	31, 1.03	16, 0,53
	51, 1.64	40, 1.11	43, 1.41	32, 1.03	16, 0.52
	61, 1.85	54, 1.34	75, 2.02	35, 1.08	27, 0.75
sbj_1	42, 1.33	45, 1.48	68, 2.17	65, 1.75	9, 0.23
	40, 1.32	58, 1.70	60, 2.04	49, 1.66	10, 0.30
	48, 1.48	51, 1.73	60, 2.08	47, 1.50	14, 0.27
	47, 1.55	56, 1.55	39, 2.08	54, 1.73	11, 0.35
sbj_2	19, 0.70	48, 1.28	36, 1.15	30, 0.97	16, 0.37
	19, 0.62	41, 1.34	37, 1.19	31, 1.00	13, 0.40
	22, 0.69	39, 1.42	36, 1.20	31, 1.03	10, 0.32
	18, 0.60	53, 1.33	36, 1.14	31, 0.99	16, 0.44
sbj_3	35, 1.08	67, 1.28	80, 2.62	40, 1.14	14, 0.43
	32, 1.09	50, 1.35	78, 2.68	41, 1.36	13, 0.41
	27, 0.96	37, 1.27	71, 2.61	42, 1.46	6, 0.364
	30, 1.03	48, 1.44	72, 2.37	37, 1.20	10, 0.35
sbj_4	31, 1.07	39, 1.33	45, 1.45	58, 2.01	13, 0.45
	30, 1.07	40, 1.14	42, 1.40	60, 1.93	10, 0.30
	35, 1.16	37, 1.30	44, 1.45	59, 1.91	13, 0.38
	29, 1.10	40, 1.18	47, 1.52	55, 1.90	10, 0.29
sbj_5	46, 1.59	46, 1.37	31, 1.03	33, 1.02	8, 0.32
	47, 1.50	44, 1.42	32, 0.95	33, 1.05	10, 0.31
	50, 1.68	35, 1.16	30, 0.98	31, 1.09	13, 0.26
	46, 1.35	48, 1.27	28, 0.99	33, 1.05	11, 0.36
sbj_6	40, 1.25	52, 1.42	83, 2.58	55, 1.63	11, 0.34
	42, 1.30	48, 1.35	78, 2.46	49, 1.63	11, 0.31
	46, 1.44	42, 1.35	81, 2.58	48, 1.55	8, 0.244
	52, 1.52	52, 1.56	87, 2.52	57, 1.75	11, 0.32
sbj_7	50, 1.58	42, 1.14	55, 1.64	31, 0.93	12, 0.35
	56, 1.76	36, 1.16	40, 1.34	27, 0.89	12, 0.31
	54, 1.79	27, 1.04	36, 1.18	28, 0.95	9, 0.30
	56, 1.87	40, 1.09	39, 1.22	31, 0.94	11, 0.34
sbj_8	23, 0.76	36, 1.34	36, 1.21	31, 0.95	11, 0.34
	29, 0.96	40, 1.32	33, 1.10	21, 0.87	10, 0.33
	25, 0.90	40, 1.28	33, 1.07	24, 0.81	10, 0.32
	53, 1.85	45, 1.46	33, 1.06	23, 0.74	9, 0.29
sbj_9	27, 1.06	48, 1.45	66, 2.34	37, 1.41	10, 0.28
	27, 1.05	48, 1.44	67, 2.37	37, 1.42	8, 0.23
	25, 0.91	49, 1.37	65, 2.59	31, 1.08	10, 0.34
	24, 0.92	48, 1.38	65, 2.60	32, 1.08	9, 0.33

of activity. Further noteworthy differences are visible in the signals of the activities *jumping, jogging_in_place* and *sit_ups*. Here, the results of the FFT suggest that *sbj_1*, *sbj_4*, and *sbj_9* altered their activity execution behavior depending on the experience gained during the study. The light-green (session 0) and blue (session 1) most dominant frequencies are more similar to each other than red (session 3) and dark-green (session 4), which in turn show greater similarities to each other than compared to the first two sessions 0 and 1. Table 3.9 provides a color-coded depiction of repetition patterns across each individual recording session. One can see that the table does not reveal any universally applicable patterns that confirm the Hawthorne Effect across all scenarios. Only two subjects, *sbj_0* and *sbj_2*, demonstrate a difference in the number of repetitions per second between monitored and unmonitored sessions. More specifically, *sbj_0* shows an increase in repetitions for 4 out of 5 activities when observed by a researcher, with only the activity *jumping_jacks* not showing such a trend. However, this activity shows an equal number of repetitions per second for both unsupervised sessions. Similarly, *sbj_2* demonstrates a higher frequency of repetitions for the activities *walking*, *jogging_in_place*, and *jumping_jacks*; yet, this behavior is even less commonly observed compared to the first scenario.

Deep Learning Analysis. Table 3.10 summarizes the average accuracy and macro F1-scores obtained during each of the three deep-learning-based experiments. Using data recorded during session 0, i.e. data originating from the same session as data used for training the network, resulted, as expected, in the highest validation metrics (70% accuracy and 64% macro F1-score).

Further, being recorded under the same conditions, validating using data recorded during session 3 resulted in the second-to-best results, being on average only around a percentage point worse than the validation using session 0 data. Surprisingly, using the self-recorded participant data, sessions 1 and 2, for validation did not result in a significant drop in performance. Even though participants recorded themselves in a completely unmonitored recording setup (session 2) performance drops were only around 4% compared to using fully-observed data. With accuracy scores being close to random guessing, Table 3.10 further shows that the shallow DeepConvLSTM [35] was incapable of being trained to differentiate data records based on the session which they originate from. Lastly, inference of networks overfitted on session 0 data showed to produce similar results across all sessions, with, though applying a different observation scenario, session 1 (semi-observed) producing the highest classification results. Overall, the results of all three experiments suggest that the predictive performance of the network of choice only marginally suffers when being used for inference on data recorded by applying a different degree of observation. Especially visualization of the per-class and per-participant results of the *(1) cross-session generalization* and *(3) fully-observed overfitting* experiments (see Figure 3.20) shows that, besides *sbj_6*, results remained stable across all sessions. Even though the non-observed setup (session 3) remained on average the least performant session, it nevertheless shows the lowest standard deviation across subjects.

Table 3.10 Average accuracy and F1-scores of the three types of performed experiments ((1), (2) and (3)). Experiments are divided by the type of session data used during validation. Results are the averages and standard deviation across subjects across three runs using three different seeds. Note that given the altered prediction scenario (session instead of activity type) experiment (2) does not involve splitting each subject's data into different session types.

Exp	Val. Set	Accuracy	F1-score
(1)	Session 0	70.11 ± 10.55	64.45 ± 12.07
	Session 1	64.46 ± 13.11	60.18 ± 13.48
	Session 2	66.02 ± 8.10	62.35 ± 8.93
	Session 3	69.33 ± 12.34	65.81 ± 12.45
(2)	All	25.21 ± 3.32	21.90 ± 3.05
(3)	Session 1	77.58 ± 11.18	76.74 ± 11.37
	Session 2	71.42 ± 8.21	69.55 ± 9.89
	Session 3	75.87 ± 9.31	74.27 ± 10.32

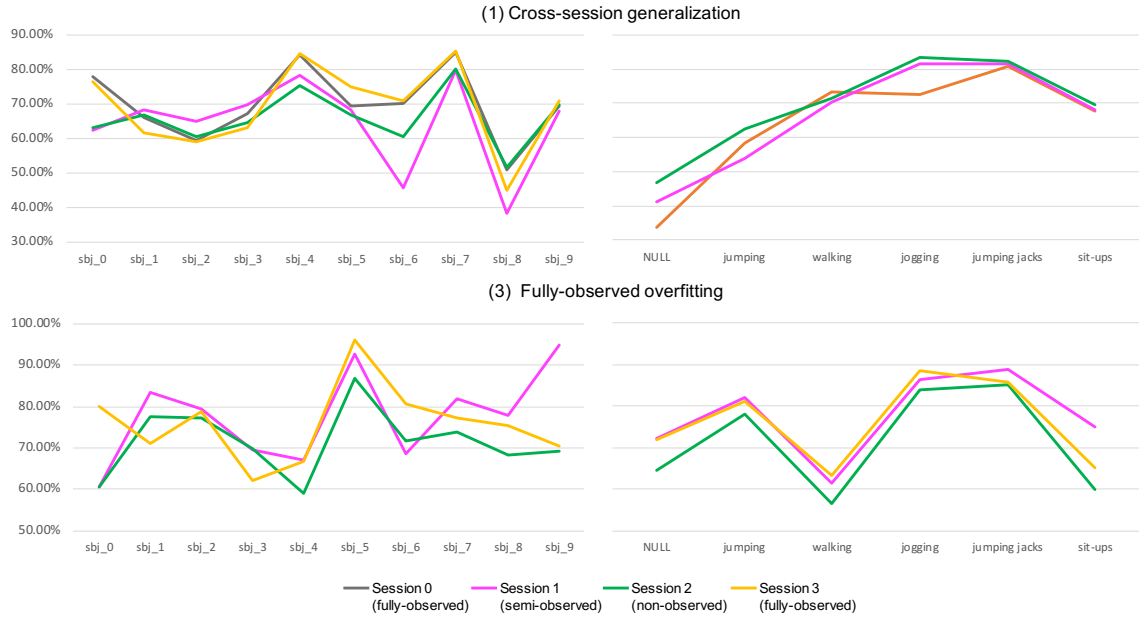


Figure 3.20 Per-subject and per-activity accuracy results of the (1) cross-session generalization and (3) fully-observed overfitting experiments. Results are averaged across three runs using three different seeds. With the exception of *sbj_6* differences amongst sessions remain marginal. Though producing the on-average lowest results, data recorded in semi- and non-observed environments were shown to be similarly applicable in terms of predictive performance than compared to fully-observed data, and, in the case of the semi-observed data, is even more reliably predicted by a network overfitted on fully-observed data.

3.4.4. Conclusions and Discussion

This paper presented a data-driven investigation aiming at measuring the effects of the Hawthorne Effect in the context of Human Activity Recognition. The study involved the recording of 10 participants performing 5 distinct activities on 4 different days. With the first and last day being supervised and video-recorded by researchers, the

remaining two days had participants self-record themselves at a location of their choice with and without video recording in place. To avoid potential biases, participants remained unaware throughout the study that the data would be analyzed to assess differences between supervised and unsupervised sessions. As part of analyzing the captured data, we employed a feature and deep learning analysis, ultimately concluding that the recorded data does not exhibit a measurable Hawthorne Effect. While the feature analysis did not reveal any generalizable patterns, the deep learning analysis showed that data originating from the unmonitored sessions produced similar classification results and even outperformed the fully observed in some cases. Although our findings do not dismiss the existence of the Hawthorne Effect, especially given the numerous clinical trials proving said effect, (see Section 3.4.1), it does challenge the prevailing notion of the applicability of laboratory compared to in-the-wild recorded data. Results of our study show that though an altered behavior of participants might be present, classification algorithms seem to learn discriminative features of similar applicability regardless.

At this point, it is important to acknowledge the limitations of this study and discuss possible reasons why the effects between the different observation scenarios were not as pronounced as we hypothesized when designing the study. Generally, the recorded dataset may not have the necessary size to draw generalizable conclusions and can only indicate a trend. Furthermore, the recorded activities solely represent a subset of periodic nature within the broader context of activity recognition. Several reasons for the lack of significant differences could be: **(1)** The participants were for all observation settings made aware that their inertial data was recorded (as this is required by the ethics council). This might mean that a possible Hawthorne Effect could have been present under all measured conditions and that this was not more pronounced when observed by additional cameras and the researchers being present. **(2)** The choice of activities could have resulted in overly simplistic movement classes that make it hard to find stark differences between the different observation sessions through our analysis methods. **(3)** It is also possible that the Hawthorne Effect in general is relatively small for our five-activity-class scenario when compared to more behavior-oriented activities (such as "brushing teeth") or fine-grained characterizations (for instance for gait analysis).

Due to the inherently limited interpretability of neural networks and the opaqueness of their decision-making processes, it is uncertain whether the observed disparities in prediction performance can be attributed solely to varying learned feature representations resulting from different levels of observation. Therefore, further investigation is warranted to explore the presence and potential impact of these influences. We believe that the results of this paper's study are nevertheless worthy of more discussion and we encourage others to perform further, more extensive research on this topic.

Section 3.5

Summary

Chapter 3 presents user-based studies focused on enhancing data recording methods and studying the human bias on dataset recordings and annotations. The research questions previously outlined have been thoroughly addressed through the studies presented in this chapter's publications.

The development of the Activate System, Section 3.1 marks a critical breakthrough, laying the foundation for many subsequent studies. Characterized by its robustness and comprehensiveness, this system enables not only our research team but also fellow scientists to collect reliable, consistent data for their studies. As an entirely open-source platform, the front-end and back-end components facilitate seamless data transfer from the smartwatch to a centralized server. With accessible source code, other users have the flexibility to customize the system to suit their specific requirements and contribute to its ongoing development. This approach promotes maximal accessibility, data sharing, and standardization of parameters across datasets recorded using this system. Moreover, the affordability of the Bangle.js 1 device streamlines procurement, allowing research labs to acquire numerous units for their projects.

In Section 3.2, a comprehensive study is presented, involving the monitoring of 11 participants equipped with a Bangle.js 1, featuring the Activate System, over a period of 2 weeks. This study is designed to address two primary questions: 1) How does the choice of annotation method impact the quality and quantity of labeled data obtained through wearable sensors? and 2) Which annotation methods are most effectively aligned with specific research objectives and use cases? To investigate these questions, participants were instructed to utilize four different annotation methods, consisting of in situ methods (Methods ① and ②) and recall methods (Methods ③ and ④).

The findings suggest that each method has its own set of advantages and drawbacks. However, in terms of precise annotations, particularly regarding time and assigned labels, in situ methods outperform self-recall methods. Conversely, self-recall methods provide a more comprehensive overview and a complete set of a participant's daily activities. It's worth noting that participants often perceived in situ methods as laborious and burdensome due to their heightened awareness and constant involvement in the study. In contrast, recall methods only require a brief 2-5 minute session for noting the day's activities along with estimated execution times. The choice between these methods should be determined by the specific research requirements. More detailed conclusions can be found in the respective section.

Multisensor synchronization is essential for datasets utilizing multiple devices, as it requires aligning separate systems with their internal clocks and maintaining this synchronization over time. In Human Activity Recognition, post-processing synchronization after data collection is more commonly used than real-time synchronization methods. However, researchers often resort to manually aligning data streams using

synchronization gestures performed at the start and end of recordings. This manual process is labor-intensive and susceptible to human error. The publication discussed in Section 3.3 addresses this challenge by introducing an automated synchronization algorithm. I propose a novel synchronization algorithm for multisensor data streams that leverages cross-correlation to identify corresponding segments in simultaneously captured sensor signals representing the same activity. By aligning these matching segments, the algorithm synchronizes the data streams. My evaluation demonstrates a median synchronization error of only 1.10 seconds, with 50% of examples achieving synchronization within 0.5 seconds precision. Furthermore, the algorithm proves effective when a distinct synchronization gesture is performed, preceded and followed by periods of inactivity. This allows for accurate gesture detection and subsequent alignment across sensor signals.

In essence, this algorithm offers a significant advancement over prior manual synchronization methods, which were both time-consuming and prone to errors. It automates the process, achieving precise alignment with demonstrably high accuracy.

In the final section of this chapter, Section 3.4, I investigate the Hawthorne Effect in data-driven HAR studies. This effect, well-established in psychological research, refers to the tendency for study participants to alter their behavior when conscious of being observed. The study records 10 participants across 4 days under 3 different conditions: (1) being observed by a researcher and filmed, (2) only being filmed while recording data at home, and (3) recording data at home without observation or filming. Through feature analysis and deep learning, the results did not reveal any clear, generalizable pattern confirming the Hawthorne Effect.

However, data from monitored sessions consistently produced similar classifiers that outperformed those trained on unmonitored data. While these findings do not dismiss the effect's existence, they suggest classifiers trained on monitored data may exhibit sufficient robustness for deployment on unmonitored real-world data.

In summary, this study provides unique insight into the potential impacts of the Hawthorne Effect in HAR research. Although no unambiguous effect was observed, the performance patterns imply that monitored data can yield robust classifiers. Further research is warranted to fully understand the implications of experimental observation on participant behavior and classifier effectiveness.

Chapter 4

Basketball Activity Recognition

This chapter applies activity recognition methodologies to sensor-based human activity data, with a focus on basketball activities captured by wrist-worn sensors. Through an initial feasibility study, I demonstrate that machine learning models like Random Forest and k-nearest neighbor can classify diverse basketball activities. This study is followed by the presentation of *Hang-Time HAR*, a comprehensive basketball activity dataset. The research questions addressed in Chapter 4 are:

(a) **To What Extent Can Basketball Activities Be Detected Using Wrist-Worn Sensors?**
   - Among the spectrum of basketball-related activities, which ones consistently demonstrate strong classification accuracy when using wrist-worn sensor data, and which activities encounter difficulties or limitations in terms of accurate identification?
   - In the context of basketball games, characterized by sporadic and dynamically changing activity patterns, can wrist-worn sensors effectively detect relevant activities? What challenges arise due to the less homogeneous nature of in-game actions?

The intersection of wrist-worn sensors and basketball activities represents a novel, high-potential area for investigation. Addressing these research questions will advance the fields of sports science and activity recognition for dynamic domains like basketball. Moreover, findings will facilitate the refinement of wearable applications in sports through sensor-based analytics.

Section 4.1

Preliminary Basketball Study

[92] Hoelzemann, Alexander, and Van Laerhoven, Kristof
Using wrist-worn activity recognition for basketball game analysis
Sept. 2018, Proceedings of the 5th International Workshop on Sensor-based Activity Recognition and Interaction
https://doi.org/10.1145/3266157.3266217

Portions of the original publication have been removed or edited for inclusion in this thesis. However, no changes were made that altered the results or conclusions presented in the original work.

Contributions:

- Both authors designed the study.
- I implemented the study, recorded and analyzed the data
- Kristof Van Laerhoven guided this work and assisted in the methodologies.

Gameplay in the sport of basketball tends to combine highly dynamic phases in which the teams strategically move across the field, with specific actions made by individual players. Analysis of basketball games usually focuses on the locations of players at particular points in the game, whereas the capture of what actions the players were performing remains underrepresented. In this paper, we present an approach that allows to monitor players' actions during a game, such as dribbling, shooting, blocking, or passing, with wrist-worn inertial sensors. In a feasibility study, inertial data from a sensor worn on the wrist were recorded during training and game sessions from three players. We illustrate that common features and classifiers are able to recognize short actions, with overall accuracy performances around 83.6%, k-Nearest-Neighbor (kNN), and 87.5% Random Forest (RF). Some actions, such as jump shots, performed well (± 95% accuracy), whereas some types of dribbling achieved low (± 44%) recall.

4.1.1. Introduction

Monitoring sports activities is a well-known field of application for human activity recognition systems, with a large number of possible use cases for recognizing and analyzing sports activities. In this paper, we introduce an approach for recognizing different kinds of activities for basketball specifically, from wrist-worn inertial sensor data (Figure 4.1). For this purpose, data has been collected during the training of a local amateur team from three participants using an IMU sensor. We annotated the gathered IMU data in five particularly challenging classes: low dribbling (ld), crossover (co), high dribbling (hd), jump shot (js), and a void class for less relevant actions, and used a supervised learning approach to examine how distinctive these motion classes are. With further development, our aim is to recognize more activities

Figure 4.1 A basketball player while dribbling and shooting (top), the raw inertial sensor data (middle plot) with classified sequences (bottom plot).

Classification. This subsection investigates the nature of the recorded raw data, the structures of the feature vectors, and the methods used in the field of machine learning. A first visual inspection shows through example data records for the data to be classified. As an example, typical patterns for low dribbling is the constant frequency of particular peaks that occur at shorter intervals in time than in the classes crossover and high dribbling, see Figure 4.2. High dribbling, as depicted in Figure

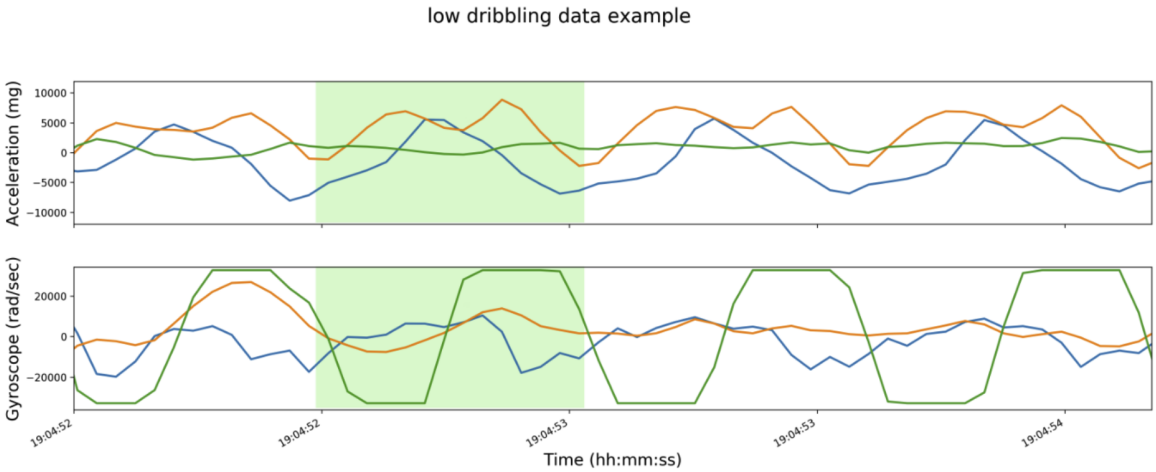


Figure 4.2 Typical time series for the low dribbling motion, showing the acceleration in milli-g and gyroscope data in rad/sec over time. Faster and more high-speed patterns can be seen in both the acceleration and gyroscope data.

4.3, can be characterized by a strongly increasing acceleration, which remains at a high level for about half a second, rather than dropping rapidly again. A crossover movement can be recognized by the fact that there is a gap of about 2 - 3 seconds between two dribblings, as seen in Figure 4.4. The reason for this is that the player wears the sensor at the dominant hand, but the ball is dribbled with the other hand for a short time, thus the acceleration on the dominant hand decreases sharply for a short period of time. In Figure 4.5 one can see the recorded data as recorded for a jump shot. Significant for this class are consecutive peaks followed by a major drop of approx. 10000 milli-g back to 0 milli-g in the acceleration. Due to the dropping acceleration of the z-axis (blue line), the first peak can be interpreted as dribbling followed by a jump shot. Based on the data we used sliding windows with a window size of one second that got classified by our algorithm. This window size has been chosen

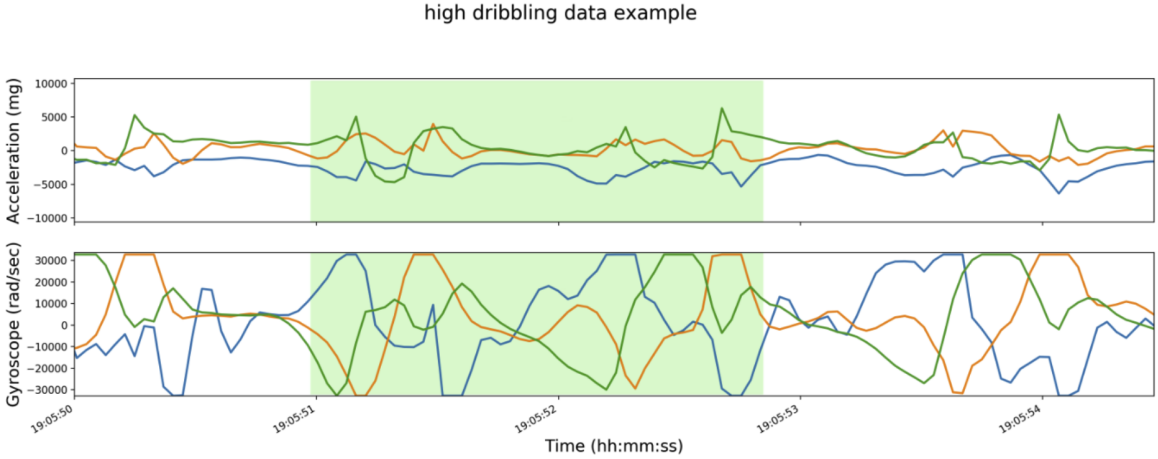


Figure 4.3 Typical time series for the high dribbling motion, showing the acceleration in milli-g and gyroscope data in rad/sec over time. Clear patterns can be seen in both acceleration and gyroscope data, but for further analysis, we will focus on accelerometer data.

because basketball is a very fast sport with rapidly changing activities. Therefore one activity mostly is in the range of milliseconds to one second. Features have been calculated for every window. For feature extraction only the acceleration data is used. In the first step, the data from the gyroscope, magnetometer, environmental sensors, or battery status are not taken into account by the algorithm. The used features are the arithmetic mean and the standard deviation for every axis of the acceleration data. This allows us to work with a 6-dimensional feature vector. In our first approach, we used a supervised learning method and focused on three different kinds of dribbling as well as jump shots. The machine learning models are trained on a small subset that contains six seconds of data per class and per participant. The sampling rate at which the data has been recorded is 25 Hz. To optimize the classification results and to improve comparability, classification has been done with a k-nearest-neighbor as well as a Random Forest classifier, both from the scikit-learn package and implemented in Python. For parameter optimization of the individual

Figure 4.4 Time series example for a crossover motion, showing the acceleration in milli-g and gyroscope data in rad/sec over time.

classifiers, we ran the cross-validation experiments mentioned in the following section for parameters over several ranges to obtain the optimal choices as listed in Table 4.1.

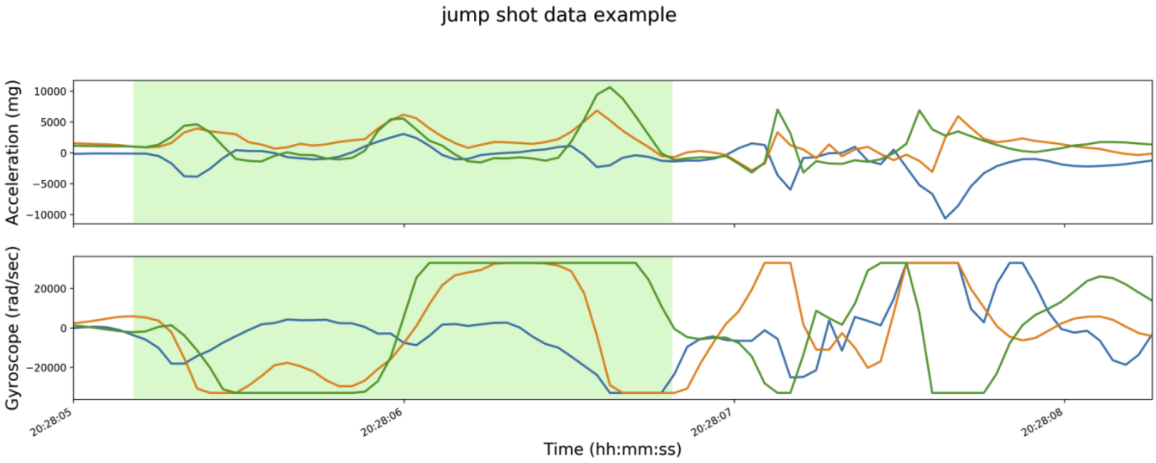


Figure 4.5 Typical time series for a jump shot, showing the acceleration in milli-g and gyroscope data in rad/sec over time. Clear patterns can be seen in both acceleration and gyroscope data.

4.1.2. The Study

In this section, the followed methodology is described in more detail and a first evaluation of the results is presented. Three participants were recruited for a user study. The participants are between 26 and 31 years old, none of them female, and all of them experienced basketball players. Participants wore the sensor, which was started approximately half an hour before the practice, on their dominant hands and were briefed on the purpose of the study before the recording. From participant

1 30 minutes or about 45000 data points were recorded, whereas from both other participants 2 and 3, one hour or about 90000 data points were recorded, summing up to approximately 235000 data points in total. For later annotation of the data, the participants were filmed during the game. With the additional time-based information from the video material, we are able to identify specific sequences in the data and annotate them with the correct label. With the labeled data we trained the model with a supervised method in combination with leave-one-out cross-validation. To avoid an imbalance in the model, we decided to limit the number of training data per class to 450 examples. To determine the accuracy, precision, and recall for each class and classifiers, their values were calculated for each iteration step of the leave-one-user-out cross-validation and finally, the average across all folds was formed. The results, as depicted in Figure 4.6, as well as the determined accuracy, precision, and recall in Table 4.1 show that it is possible to achieve an average accuracy of 87.6% even with few training data and simple features. The confusion matrices show that the classes

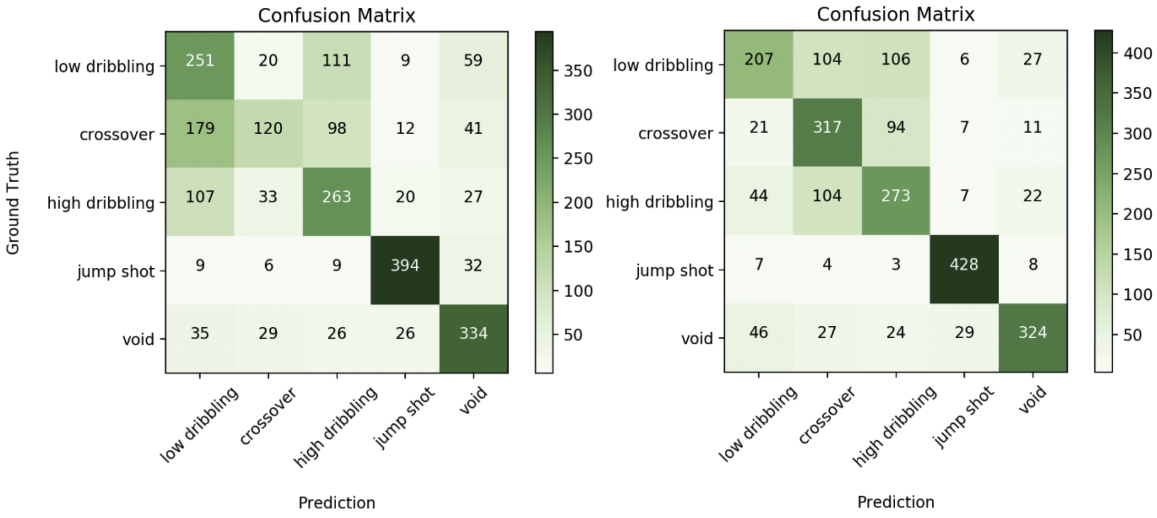


Figure 4.6 The confusion matrix for kNN (left) shows particular confusion among dribbling and good results for jump shots. The confusion matrix for Random Forest (right) shows a slightly better per-class performance and equal confusion among the different dribbling actions.

low dribbling and high dribbling are slightly better recognized than crossover. Above-average recognizable are jump shots. The recognition of this class is already possible with an accuracy of 96,6%. Due to the good accuracy, but fluctuating precision, it can be stated that the values of the features formed vary greatly, but the classification is nevertheless largely correct. This suggests that in the further course of research, the annotation of the training data must be carefully examined again and, if necessary, improved. Both classifiers vary in their results in terms of precision and recall. This leads to the conclusion that in future works more classifiers should be tested with our data. An average accuracy of 87.5% is not yet optimal. This still leaves room for improvement of the system. An extension or refinement of the used features would result in an improvement of the algorithm. The presented application setup shows a

Table 4.1 The accuracy, precision, and recall performance in percent for all classes: fast dribbling (fd), crossover (co), high dribbling (hd), jump shot (js), and background data (void), together with the best-performing parameters per classifier.

	k-Nearest-Neighbor				
Class	**ld**	**co**	**hd**	**js**	**void**
Accuracy	76.5	81.4	81.0	94.5	87.7
Precision	43.2	57.7	51.8	85.5	67.7
Recall	56.0	27.0	58.4	87.6	74.2
F1-Score	49.0	37.0	55.0	87.0	71.0
Hyperparameters	n_neighbors=4; leaf_size=10; algorithm=kd_tree; metric=minkows; weights=distance; p=1				
	Random Forest				
Class	**ld**	**co**	**hd**	**js**	**void**
Accuracy	83.5	83.0	82.4	96.6	91.9
Precision	62.4	56.3	55.5	89.6	82.0
Recall	44.2	67.8	61.5	94.2	76.0
F1-Score	52.0	62.0	58.0	92.0	79.0
Hyperparameters	n_estimators=10; max_features=auto; min_sample_leaf=1; min_sample_split=2; max_depth=None; bootstrap=true				

novel combination that works well under laboratory conditions and with hardware comparable to the design and comfort of smartwatches.

4.1.3. Discussion

In this section, the most related work of [155] is compared to our proposed method. Furthermore, the differences, as well as the advantages and disadvantages of both approaches will be discussed. The technical setup of [155] consists of five self-developed boards with installed IMU sensors. The hardware needs to be placed on the player's body. One needs to be attached at the lower back, and each one on both legs and feet. Those five devices are recording the data independently from each other. The recorded data run through the common known processes of a machine learning application, i.e. preprocessing, segmentation, feature extraction, and classification. For recognizing a specific activity a decision tree has been developed. Wherein, the first step is to distinguish between a standing and moving activity. Only after a moving activity has been recognized they decide which movement has been executed. Ten features are utilized to calculate the correct class, for every segment of data each accelerometer, in a total of four values, is obtained and transformed into a feature vector. The used features are range, sum, mean, standard deviation, mean crossing rate, skewness, kurtosis, frequency bands, energy, and number of peaks above a threshold.

The sampling frequency that is used was first set at 200 Hz but was down-sampled to 40 Hz for the accelerometers due to redundant data. In contrast to this, the approach presented comes with a single IMU worn at the wrist, which is the most active part of the body while playing basketball for the player who currently owns the ball. As a result of this, the focus of our system is set on the direct interaction with

the ball. The used features are limited to mean and standard deviation for every axis of the accelerometer. Therefore the setup is less complex compared to [155]. Both approaches are evaluated with the signals of three participants. The lower complexity of the experiment is at the same time its greatest advantage. The small number of devices involved results in less redundant data. In addition, the system offers less space for disturbing factors. Furthermore, with only one device that the player has to wear on the body, the system offers better wearing comfort and has less impact on the player's performance. As other works already depicted, for example, [53], it is also possible to detect walking or running activities by only wearing one wrist-placed IMU sensor. Due to these circumstances, we would prefer a test setup as proposed by us and try to improve it further to be able to classify more activities and improve the accuracy.

4.1.4. Conclusions

We argue in this work that wristwatch-based motion sensors are ideally placed to detect basketball-relevant actions and gestures. The results of this first feasibility study suggest that it is possible to classify different movements of a basketball player using an inertial sensor that is worn on the wrist. Through this feasibility study, it is now possible to expand the system and add more activity classes. By completing the system and the resulting possibility to recognize all actions of a basketball game only by means of acceleration data, it is possible in the following to recognize the actions of players in real-time and without the help of video data annotation. This enables a live analysis system that is able to visually display the recorded games and, in the next step, develop the system for live game analysis. Furthermore, one could use the system for training purposes and thus design, for example, a feedback system that gives the training player feedback as to whether the action he was currently performing was technically correct. This would be especially useful for shooting training: The board could be equipped with visual feedback reflecting the correctness of the action performed, or offer more detailed action analysis.

Section 4.2

Hang-Time HAR

[92] Hoelzemann, Alexander et al.
Hang-Time HAR: A Benchmark Dataset for Basketball Activity Recognition Using Wrist-Worn Inertial Sensors
June 2023, MDPI Sensors 2023, 23(13), 5879, Special Issue "Inertial Measurement Units in Sport"
https://doi.org/10.3390/s23135879

Portions of the original publication have been removed or edited for inclusion in this thesis. However, no changes were made that altered the results or conclusions presented in the original work.

Contributions:

- All authors contributed to the conceptualization of the study.
- Julia Lee Romero and I recorded and annotated the dataset.
- I implemented the feature analysis and the corresponding visualizations.
- Marius Bock implemented the Deep Learning Analysis and the corresponding visualizations.
- Kristof Van Laerhoven and Qin Lv have guided this work and assisted in the methodologies.

This section presents a benchmark dataset for evaluating physical human activity recognition methods from wrist-worn sensors, for the specific setting of basketball training, drills, and games. Basketball activities lend themselves well for measurement by wrist-worn inertial sensors, and systems that are able to detect such sport-relevant activities could be used in applications of game analysis, guided training, and personal physical activity tracking. The dataset was recorded from two teams in separate countries, the United States of America (USA) and Germany, with a total of 24 players who wore an inertial sensor on their wrist, during both a repetitive basketball training session and a game. Particular features of this dataset include an inherent variance through cultural differences in game rules and styles as the data was recorded in two countries, as well as different sports skill levels since the participants were heterogeneous in terms of prior basketball experience. We illustrate the dataset's features in several time-series analyses and report on a baseline classification performance study with two state-of-the-art deep learning architectures.

4.2.1. Introduction

Human activity recognition (HAR) systems aim to track people's physical movements and categorize them according to predefined activity classes or clusters. Methods from machine learning, and especially deep learning, are applied in order to classify

samples of sensor data into predefined classes. According to [61], only 30 datasets in total have ever been released publicly and 11 out of the 13 most cited datasets in the HAR community were released in 2015 or prior. Such datasets, especially the older ones, are commonly recorded in laboratories and follow strict activity protocols and movement patterns. Since scientists lack solid annotation methods and tools for recordings in-the-wild, they tend to fall back to a controlled lab environment, in which visual systems, often cameras, can be installed to facilitate labeling the sensor data in hindsight. Due to the labor-intensive work of labeling data, the number of participants is often limited. Significant hurdles for experiments conducted in-the-wild lead to an imbalance in the number of publicly available datasets from controlled environments in comparison to uncontrolled environments. However, depending on the design of the experiment itself, a sports environment, e.g. Figure 4.7, can be seen as a semi-controlled environment, since its recording sessions can include both practice drills (controlled) and game sessions (uncontrolled). Due to the nature of the sports domain, this data contains highly variable and dynamical movement patterns, which exhibit high intraclass variability, as well as high intersubject variability [142] due to gender, height, weight, personal play style, and athletic ability of the subject. These differences are important in real-world scenarios, and classifiers, in general, perform worse on in-the-wild datasets than on lab-recorded datasets due to effects

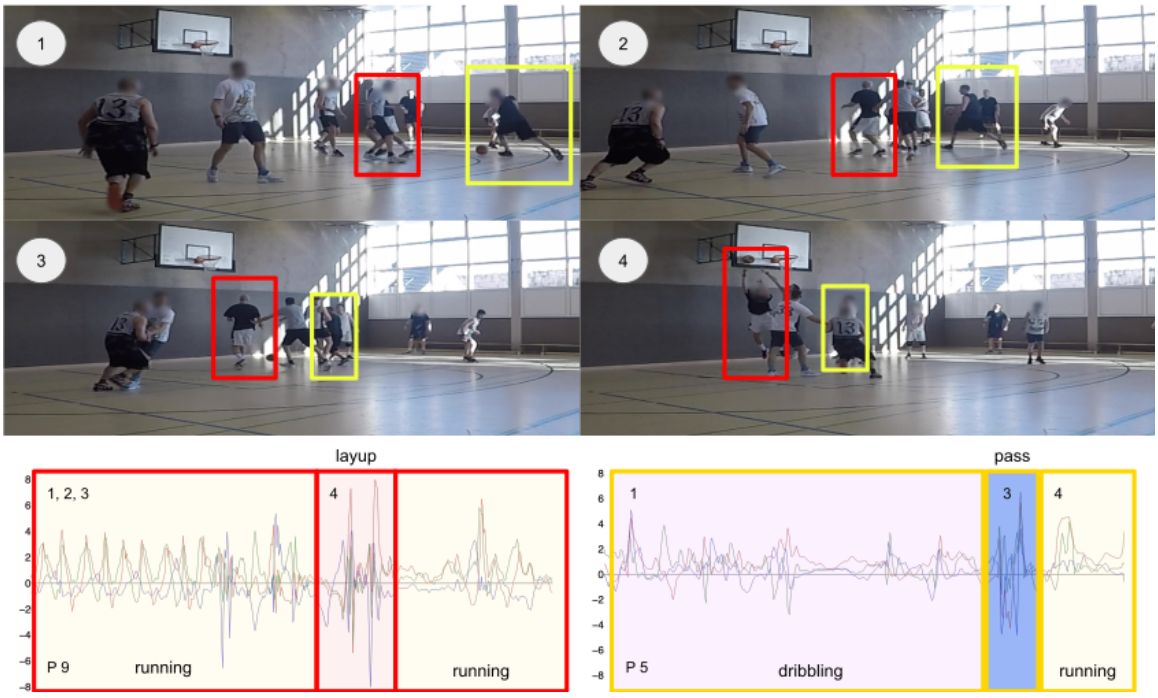


Figure 4.7 A scene and activities from the dataset: Offensive play of player 12 (yellow) and player 6 (red), see Table 6, with player 12 dribbling the ball (1), (2), and then passing (3) to player 6. Player 6 then performs a layup (4). Video frames 1–4 and the performed activities are highlighted in the time-series below. The activity *running* is marked as yellow, *layup* as red, *dribbling* as mauve and *pass* is colored in blue.

such as weak labeling [154] or smoothness in the performed activities [69]. Therefore, providing a publicly available dataset containing complex sports-related activities can have an important impact on how we design and validate our future HAR algorithms and gives researchers the security of a semi-controlled environment with precise labels based on video recordings.

Many previous studies, as summarized in Table 2 (for sports), Table 3 (for basketball), and earlier published surveys (e.g., [168, 51]), demonstrate that there is an interest in using inertial measurement unit (IMU)-based wearable solutions for activity recognition in sports. Professional athletes use sensor-based training methods to improve their sporting skills. The German professional soccer clubs, Hannover 96 and 1. FC Magdeburg utilizes the commercial body-worn IMU sensors, Vmaxpro [31], which monitors the athletes' movements and presents training recommendations, including specific strength training exercises, via smartphone for the trainer and athlete. In 2021, the sports fashion company *Adidas* released a sensor-equipped inlay sole for a soccer shoe [7], which is capable of detecting soccer-specific activities. Similarly, in 2020 the Finnish-based company SIQ [189] released a sensor-equipped basketball with feedback aiming to improve players' shooting skills.

Contributions: This dataset is the first publicly available dataset with sensor-based basketball activities collected from teams of players doing both structured practice drills and an unstructured game. The classes included were selected by researchers with many years of experience in playing basketball and represent a full range of basketball activities that cover key aspects of the sport. The activities included show high dynamics, complexity, and variability within the same subject (due to different execution styles) and also between subjects (due to experience and play style). Since recordings are split into warm-up, drill, and game sessions, the dataset provides a mix of controlled and uncontrolled environments. The game shows a higher dynamic because of the influence of other players and a higher pace than that in the drills. This setup can be seen as a transition from a controlled environment to a semi-controlled recording environment. Because the dataset does not contain information about successful scoring, it is not necessarily meant to be used for skill assessment. However, the metadata does contain information about the players' experience (novice or expert). Novices are players with little prior experience in playing basketball. Players' execution of activities can therefore be expected to display a large variance. Since the dataset was recorded from 24 participants (roughly the size of two complete teams) across two continents, it also includes the inherent differences within the rule sets played by the International Basketball Federation, franz. Fédération Internationale de Basketball (FIBA), in Europe and the National Basketball Association (NBA) in Northern America. This is a unique setup that is not available in other sports.

Impact: This dataset can be used by the Ubiquitous Computing community to tackle a variety of research questions in the area of Human Activity Recognition. A (sports) dataset of this scope and study design is not yet publicly available, since it contains the same set of activities recorded with the same hardware and sensor modalities in both controlled and uncontrolled environments. The study is multi-

part where some parts are controlled by prescribed activities and other parts are uncontrolled, such as a "free-movement" game. The nature of basketball is such that this mix of controlled conditions is easily captured in video and manually labeled in detail. The multi-part study adds complexity and diversity to the data and gives researchers a new playground to benchmark algorithms and approaches as well as (possibly) spot deficiencies in existing state-of-the-art architectures. Furthermore, the game phases include data that models team and game dynamics. This feature is something that can be explored more deeply by future studies with regard to group activity recognition. This work provides layered labels, or multi-labels, as certain activities consist of a series of other activities. The complex characteristics of the data and the differences due to the location can help to address open research problems, such as Transfer Learning and Data Augmentation, recognizing complex activities in dynamic and real-world environments. Further development in these areas may include - but is not restricted to - research focused on data recording techniques and annotation procedures, data preprocessing (e.g., data segmentation), feature extraction, or developing new deep learning methodologies and evaluation methods. Methodology-wise, we restricted our recording setup to commercial and mostly open-source recording components (in particular the smartwatches and their firmware). Such a low-effort recording setup has the significant advantage of being deployable in spontaneous situations and would not be restricted to basketball - if further developed. The labeling setup builds on previous work by the community and focuses on reproducibility by other research labs.

4.2.2. Motivation

Basketball is played across the globe, but the two of the most dominant rule sets are (1) Fédération Internationale de Basketball (FIBA) [70], which is played by: Basketball Champions League, Euroleague Women, Basketball Champions League Americas, FIBA Europe Cup, EuroCup Women, FIBA Asia Champions Cup, FIBA Intercontinental Cup, Olympic Games - and, the most important basketball league worldwide - (2) National Basketball Association (NBA) [153] - played in North America. The two sets of rules are similar but differ in several details[1]. For example, in contrast to the FIBA rules, NBA rules allow players to do a so-called 0-step - an additional step between catching the ball and the first dribble. Other differences include game time (40 minutes for FIBA games vs. 48 minutes for NBA games) and basketball court dimensions (28m x 15m for FIBA vs. 28.7m x 15.2m for NBA). In addition to these differences, the play styles in professional European and North American basketball tend to be slightly different as well. NBA teams often build their game around one or a few star players and a more aggressive defense. In contrast, European teams focus more on team play and a compact defense.

Basketball is a very dynamic and highly intense sport that combines fast movements, quick switching between offense and defense, and diverse execution of activities.

[1] https://www.fiba.basketball/rule-differences

Activities in this sport can be characterized into one of the following activity categories (1) short actions or micro-activities (passing or rebounding), (2) complex activities (shooting the ball, layups), and (3) periodical activities (sitting, standing, walking, running and dribbling). These activities are performed differently by every player, but also by the same player depending on factors such as in-game situations, physical fatigue and stress level, mental state, and improving skills over time. For that reason, the three research challenges defined in 2014 by Bulling *et al.* [45] – (1) intraclass variability, (2) interclass similarity, and (3) the NULL-class problem – are all reflected by the dataset presented here. These three challenges become more significant with less structured, real-world data, such as data from a real basketball game. In a game situation where external influences, such as other players, affect the gameplay and physical movements, these characteristics are more apparent.

A dataset like the one presented in this article, which is recorded during two real basketball training sessions lasting between 1 - 2 hours, also offers the opportunity to close the gap between controlled and uncontrolled study setups. After a few minutes in the warm-up session, participants reported that they forgot that they were monitored through their smartwatches and nearby cameras, and behaved like they would in a usual training session. We argue that the basketball game part of the dataset equally encouraged participants to move naturally [69], [14]. Results show that even though science is advancing fast in the area of HAR, it is still challenging to train machine learning models that are capable of reliably detecting activities in naturalistic scenarios, such as [26]. In order to overcome this challenge, we consider the next important step in HAR to be that future algorithms are developed and evaluated on realistic datasets. Sports HAR datasets in general can be the perfect setting for researchers to do exactly this and they could allow for deeper insights for sports scientists as well as the deep learning and HAR community. Specific datasets that contain sports activities, e.g., DSADS [11] or the study presented by Trost *et al.* [208], often contain a variety of different sports in one dataset and reduce entire sports, such as playing basketball, to single target classes to be detected. The UTD Multimodal Human Action Dataset [48] contains four repetitions from eight subjects of 27 different activities from a variety of domains, such as sports. However, the included sports activities are limited to one specific activity per sport, e.g., shooting a basketball. Activities from these datasets are not representative of an entire sport. Inertial sensor-based and sports-specific datasets that capture the variability and complexity of a sport are not yet available in public repositories. Even recently published datasets, such as TNDA-HAR [229], focus on simple periodical locomotion activities, and additionally, available datasets that are used by the Ubiquitous and Pervasive Computing community rarely combine (1) scope, (2) quality, (3) variability, (4) complexity, and (5) reproducibility in the same benchmark dataset. The basketball data published by [11, 208, 48] does not represent the same level of complexity regarding the recently mentioned characteristics with the dataset we present since the class defined as *basketball* is highly simplified. We, therefore, highlight this as one of the main motivations for such a dataset. The following Table 4.2 gives an overview of relevant HAR datasets.

As one can see in the Environment column, most of the datasets are recorded in controlled environments, partly because data collection is easier, and partly due to the lack of annotation methods without synchronized video recordings. Tools such as [137], [157] or [166] are designed to be used in hindsight, with video footage and with well-defined synchronization gestures at the beginning and end of the video. Among the datasets presented in Table 4.2, only ActiveMiles [170] and Leisure Activities [26] are recorded in an uncontrolled environment. ActiveMiles is limited to simple

Table 4.2 The most relevant datasets used by the HAR community, as well as examples for datasets from uncontrolled or semi-controlled environments (with challenges based on Table 1 from [51]).

Dataset	Device	# Subjects	# Classes	Domain	Environment	Challenges	Published
HHAR [195]	Smartphone	9	6	Locomotion	Controlled (Lab)	Multimodal, Distribution Discrepancy	2015
RWHAR [198]	Smartphone, Wearable IMUs	15	8	Locomotion	Controlled (Outside)	Multimodal	2016
Opportunity [177]	Wearable IMUs, Object-Attached Sensors, Ambient Sensors	4	9	ADL, Kitchen Activities	Controlled (Lab)	Multimodal Composite Activity	2010
Opportunity++ [55]	Wearable IMUs, Object Attached Sensors, Ambient Sensors	4	18	ADL, Kitchen Activitie, Video, OpenPose tracks	Controlled (Lab)	Multimodal Composite Activity	2021
PAMAP2 [173]	Wearable IMUs	9	18	Locomotion, ADL	Controlled (Lab, Household)	Multimodal	2012
Skoda [237]	Wearable IMUs	1	12	Industrial Manufacturing	Controlled (Industrial Manufacturing)	Multimodal	2008
UCI-HAR [12]	Smartphone	30	6	Locomotion	Controlled (Lab)	Multimodal	2013
WISDM [111]	Wearable IMUs	29	6	Locomotion	Controlled (Lab)	Class Imbalance	2011
UTD-MHAD [48]	Wearable IMUs, Video	8	27	Gestures, Sports	Controlled (Lab)	Multimodal	2015
Daphnet [18]	Accelerometer	10	3	ADL, Locomotion	Controlled (Lab)	Simple	2009
DSADS [11]	Wearable IMUs	8	19	Sports, ADL	Controlled (Lab & Gym)	Multimodal	2010
ActiveMiles [170]	Smartphone	10	7	Locomotion	Uncontrolled (In-The-Wild)	Real-World	2016
Baños *et al.* [20]	Wearable IMUs	17	33	Sports (Gym)	Controlled (Gym)	Multimodal	2012
Leisure Activities [26]	Wearable IMU	6	6	ADL	Uncontrolled (In-The-Wild)	1 activity per subject	2012
WetLab [181]	Wearable IMU, Egocentric Video	22	9	Experiments (Wetlab)	Semi-Controlled (Wetlab)	Multimodal	2015
TNDA-HAR [229]	Wearable IMUs	23	8	Locomotion	Controlled (Lab)	Multimodal	2021
CSL-SHARE [122]	Wearable IMUs, EMG, Electrogoniometer, Microphone	20	22	Locomotion, Sports	Controlled (Lab)	Multimodal	2021
Hang-Time HAR	**Wrist-worn accelerometer**	**24**	**15**	**Sports (Basketball)**	**Controlled and uncontrolled (Gym)**	**Different recording environments, Class Imbalance**	2023

locomotion activities, and Leisure Activities consist of six participants' wrist-worn inertial data over a week where each of them performed one specific leisure activity daily. The WetLab dataset can be seen as recorded in a semi-controlled environment, where participants were told to follow a specific protocol for an experiment in the wet lab, but they were allowed to execute steps in their preferred order and at their own speed. This environment in combination with the sporadic activities makes it a difficult dataset to learn for machine or deep learning models with results of ~40% F1-Score. Stoeve *et al.* [196] took IMU-based activity recognition from the lab to a real-world soccer scenario where passing and shooting in real soccer games are recognized. This study did not publish the dataset publicly, however. We consider

sports in general to be a highly interesting scenario for benchmark datasets that are aimed at further developing learning mechanisms that are also capable of detecting periodic activities, such as *sitting*, *standing*, *walking*, *running*, short or micro-activities such as *passing*, or *rebounding* and also complex activities such as *shooting* a basketball or performing a *layup*.

Finally, we summarize the main features of our Hang-Time HAR dataset as the following:

(a) Hang-Time HAR consists of wrist-worn inertial data from 24 participants from two teams and from two countries with two different rule sets, performing 10 different basketball activities.

(b) Hang-Time HAR is recorded in three different types of sessions: (1) warm-up, (2) drill, and (3) game. The drill sessions are executed in a structured way where participants were instructed to execute single specific activities, in a predefined order. However, the warm-up and game session followed the teams' typical routine and were not tied to an activity protocol and participants were allowed to play as they preferred.

(c) Hang-Time HAR includes considerable variety, with both simple and periodic activities, short or micro-activities, and complex activities. Hang-Time HAR also explicitly contains data from participants with different experience levels and following different basketball rule sets.

(d) Hang-Time HAR is labeled on four different layers: (I) coarse, (II) basketball, (III) locomotion, and (IV) in/out. This will allow future researchers to combine labels, such as for example *dribbling + walking*, *dribbling + running*, or *jump shots*. This results in more complex activities and it becomes more challenging for the classifier to perform well.

4.2.3. Methodology

This section provides detailed information about the study parameters, the hardware, and software used to record the data, the preprocessing and labeling process, as well as recommendations for other researchers for recording IMU data. The second part of this section describes in detail the dataset in regard to the class characteristics.

Study Design. This dataset contains data collected during two separate periods and following the same study protocol. The first author supervised Study 1 at the University of Siegen, following FIBA regulations, which did not require Institutional Review Board (IRB) review. The second author conducted Study 2 at the University of Colorado Boulder, according to NBA regulations, and the study is IRB-approved. In subject recruitment, we excluded any person with a disability impairing their ability to play basketball and any person under the age of 18 years. Four modes of data were collected during the study: information collected manually by researchers, online questionnaires, smartwatch accelerometers, and video cameras in order to annotate the accelerometer data. However, the video data contains information that could

de-anonymize our participants and is therefore not included in the dataset.

Prior to the study, participants signed a consent form that outlined the study protocol and risks of harm, and they were informed that the questionnaire and accelerometer data will lack any personally identifiable information and that a dataset containing these two modes of collected data will be made publicly available. At the start of the study, participants received one smartwatch and were assigned a unique identifier. The researchers manually collected the unique ID and name of the participants, in order to allow them to retroactively request for their data to be deleted prior to the release of the dataset. Participants filled out an online questionnaire collecting age, height, weight, gender, dominant hand, and history of playing basketball. Participants were then instructed to wear the smartwatch on their dominant hand and perform a sequence of basketball-related activities (i.e., standing, walking, running, dribbling, shooting, layups, and a game). Two cameras were used to record each study, see Figure 4.8, and the footage was combined for the labeling process. The study protocol

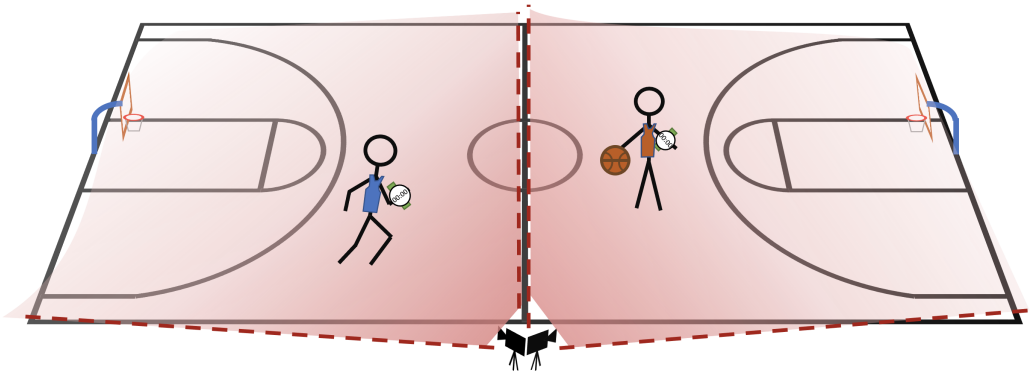


Figure 4.8 Our study design used 24 subjects with 13 subjects living in Germany and 11 subjects living in the United States of America. In each study, the players simultaneously performed the drills and game while the entire basketball court was monitored using two wide-angle cameras. After the study, the camera footage was used for detailed annotation of all activity-relevant data.

is divided into two parts. The first part is designed to collect controlled data by having participants complete a sequence of predefined activities for a defined period of time, while this first part is controlled, it also simulates real-world basketball drills in practice sessions where players repeatedly practice a certain activity (e.g., layups, shooting, dribbling, running). The second part is a basketball game between two teams each with five players per team on the court, and extra players rotated into the game. Video cameras were set up along the sidelines of the court in order to record each participant's activities for the labeling process. The differences between the two studies as well as the specifications of the recorded videos are documented in Table 4.3. We have several recommendations for the collection of similar datasets based on our own experiences conducting the study and annotating the data. In the context of this study, we recommend setting up two wide-angle lens cameras, e.g., GoPro, side-by-side with each one capturing one half of the court and additionally instructing

Table 4.3 Differences between the two studies and a description of the camera recording settings and file sizes for each study and camera employed.

	Ball Regulation	Number of Participants	Study Duration	Video Camera	Duration (Minutes)	Resolution (Pixels)	File Size	FPS	SD Card Capacity
Europe	FIBA	13	110	GoPro Hero 4	110	1920 x 1080	20 GB	60	64 GB
				GoPro Hero 8	110	1920 x 1080	20 GB	60	64 GB
North America	NBA	11	76	GoPro Hero 8	76	2704 x 1520	26 GB	60	125 GB
				Sony NEX6	40	1920 x 1080	5 GB	60	32 GB

participants to wear uniquely colored clothing to aid the labeling process. We found that the cameras have a short battery life and it was necessary to bring extra batteries or a portable power bank to continuously charge the camera for the duration of the study. Finally, in order to synchronize the video footage with the smartwatch data, we recommend having participants complete a synchronization gesture, such as jumping, simultaneously on video at the start and end of the study.

Hardware: Each subject's inertial data was captured by an open-source smartwatch, which was fitted to the user by the author conducting the study to fit comfortably around the dominant wrist. This watch was used to record 3D accelerometer data at ~50 Hz and at a sensitivity of ±8 g, This watch was used to record 3D accelerometer data at ~50 Hz and at a sensitivity of ±8 g, using the Bangle.js smartwatch with our custom firmware [211] The watch firmware was programmed to record the acceleration data and display the current time and date. It did not need pairing to other (e.g., Bluetooth) devices during the study. The axis orientation, viewed from above, is as follows: +X-axis points at a 90° angle to the left, +Y-axis points at a 90° angle forward and the +Z-axis points upwards at a 90° angle.

Controlling the Bangle.js Smartwatches: The Bangle.js smartwatches can be controlled with a custom smartphone app, which is implemented as an open-source cross-platform solution using Flutter [79] and is made available on the Apple AppStore, the Google Play Store, and on Github [209]. The app communicates via Bluetooth Low Energy with smartwatches. In order to download the data from the devices after stopping the recordings, the smartwatches can be connected via Web-BLE with a local PC. Through a website [210] the devices' flash storage space can be accessed. The following Figure 4.9 depicts the procedure of starting the smartwatches. The first screen of the figure shows the app searching for nearby Bangle.js devices. After all nearby devices were found, 4 smartwatches in total, one can either start all devices individually or press the button "Start All" to start all visible devices simultaneously, screen (2). Both options open a dialogue, screen (3), where the researcher can choose the sampling rate (Hz), sensitivity (g), and starting time. Available sampling rates are 12.5, 25, 50, and 100 Hz and the sensitivity can be set to ±2, ±4, and ±8 g. The smartwatches need to be programmed to start at a preselected full hour. If the device should start immediately it needs to be set to the current hour. After pressing the Start button (3), the app connects to either one or all Bangle.js devices, synchronizes the time, and programs the preselected parameters. We did not evaluate how many Bangle.js smartwatches can be started simultaneously; however, we did not encounter any issues while starting up the 14 devices at the same time.

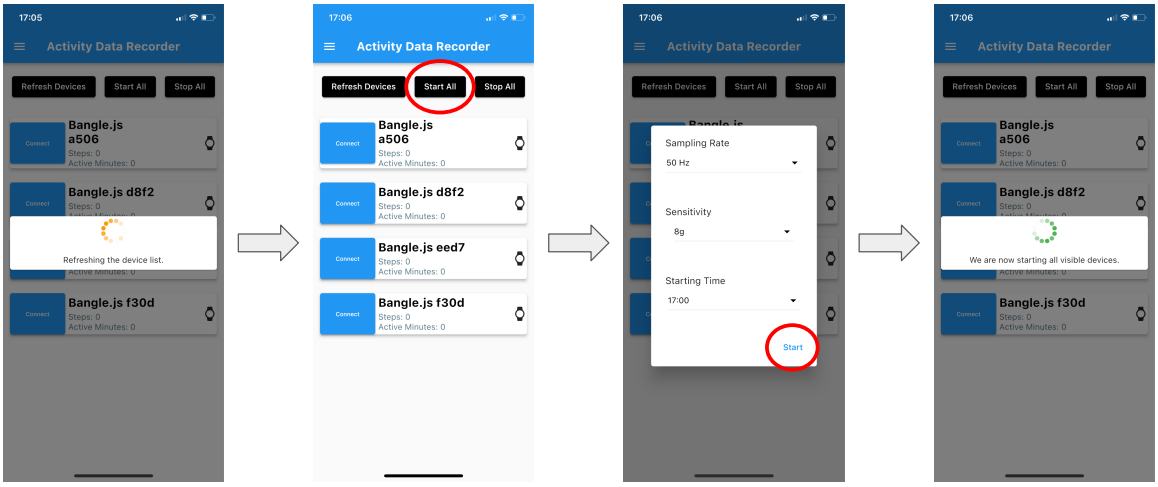


Figure 4.9 Our custom smartphone app was used to synchronize all smartwatches' real-time clocks at the beginning of each recording through Bluetooth Low Energy (BLE) serial commands and start recording simultaneously. After the app is started, it first scans for all available Bangle.js smartwatches. After that, the user has the option of either starting all devices simultaneously or individually. Before the smartwatches are started, the user is asked to enter the desired parameters (sampling rate, sensitivity, and start time). After pressing the start button, all smartwatches are started with the desired parameters.

Obtaining Ground Truth. The raw accelerometer data is stored in CSV format. The labeling of ground truth was performed in hindsight with the multimedia annotation tool EUDICO Linguistic Annotator (ELAN) [43], which was originally developed as a linguistic annotation tool. The tool has the functionality to visualize additional time-series data [202] and display both modalities together. Before the annotation of

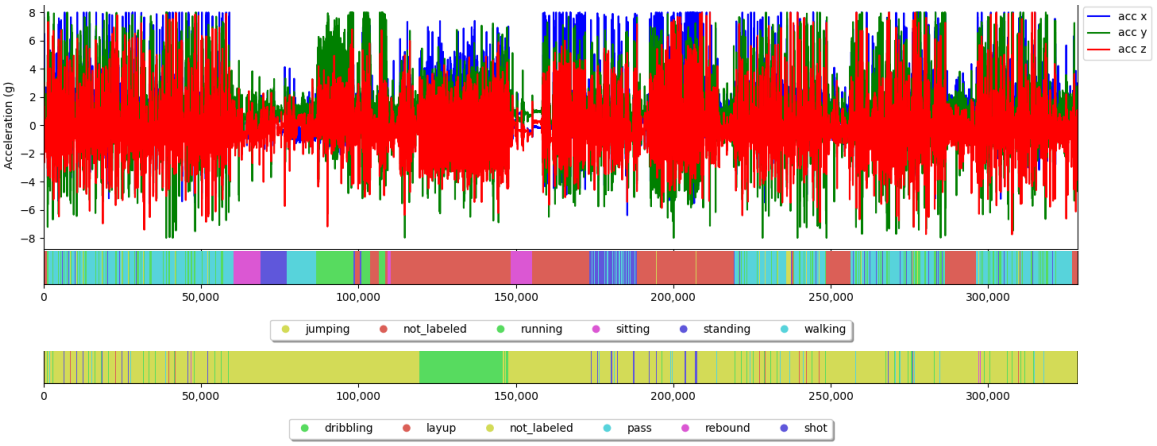


Figure 4.10 Illustration of the multi-tier labeling approach, depicting the inertial data of subject *05d8_eu* (top), the ground truth locomotion Layer (middle), and the ground truth basketball layer (bottom).

the data, we ensured that the sensor and video data were aligned with each other by using a jumping as a synchronization gesture with a few seconds of sedentary activity before and after the jump. The accelerometer data were then manually moved to the correct position. Figure 4.10 shows exemplary ground truth for Locomotion and the Basketball layer of subject 8 (with ID *05d8_eu*).

Most of the samples are labeled as *not_labeled*, especially on the basketball layer, since basketball activities tend to occur sporadically (whenever a player has the ball).

Dataset. The term Hang-Time generally refers to the time a player spends in the air while shooting or passing a ball. This term, however, has been used by sports magazines [190], game developers [225] or producers of basketball equipment [28] as an inspiration to name their product. We decided to name our dataset Hang-Time HAR - which is focused on the time-series analysis of basketball activities - due to its high memorability and its short and succinct form. The name represents to us the dataset's direct relationship between basketball, **time**-series data classification and therefore human activity recognition. The name was consensually approved by the authors. Hang-Time HAR provides accelerometer data recorded with ~50 Hz and ±8 g. Even though a full IMU has not been used, the data provided can be specified as complex due to the given classes. Table 4.4 provides additional meta-information about every participant. The same information is available in the file *meta.txt* and downloadable from the dataset repository. In total, we recorded ~1:50:00 h of 13 participants from Germany and ~1:16:00 h of 11 participants from the USA. The study was conducted

Table 4.4 Meta information as given through the study questionnaire by all participants, 13 from Germany, Europe (eu) and 11 from USA, North America (na). A total of 3 participants were female and 21 were male. The players were between 18 and 39 years old. Through self-assessment, in which participants were asked to evaluate their experience in basketball, 8 players responded with novice and 16 with expert. Two people were left-handed. Additional about the anthropomorphy of our participants are excluded due to restrictions given by the Ethical Council of our university. **Note:** Subject 2dd9_na wore the smartwatch on the left wrist even though the right hand is dominant.

Europe													
#	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**	**9.**	**10.**	**11.**	**12.**	**13.**
ID	**e90f_eu**	**b512_eu**	**f2ad_eu**	**4991_eu**	**9bd4_eu**	**2dd9_eu**	**ac59_eu**	**05d8_eu**	**a0da_eu**	**10f0_eu**	**0846_eu**	**4d70_eu**	**ce9d_eu**
Age	25	39	20	28	19	34	29	19	20	35	18	36	25
Dom. Hand	right	right	left	right	left	right	right	right	right	right	right	right	right
Height (cm)	191	167	178	188	190	196	190	178	193	172	171	188	175
Weight (kg)	85	85	67	100	80	83	83	77	87	773	60	74	73
Gender	male	male	male	male	male	male	male	male	male	male	male	male	male
Experience	expert	expert	expert	expert	expert	expert	expert	expert	expert	expert	novice	expert	expert
North America													
#	**14.**	**15.**	**16.**	**17.**	**18.**	**19.**	**20.**	**21.**	**22.**	**23.**	**24**		
ID	**b512_na**	**9bd4_na**	**2dd9_na**	**4d70_na**	**c6f3_na**	**f2ad_na**	**a0da_na**	**ac59_na**	**10f0_na**	**0846_na**	**ce9d_na**		
Age	27	26	24	26	24	25	28	28	27	30	24		
Dom. Hand	right	right	right	right	right	right	right	right	right	right	right		
Height (cm)	165	178	175	183	180	170	170	173	154	165	188		
Weight (kg)	68	65	84	68	83	69	73	65	49	65	73		
Gender	male	male	female	male	male	male	male	male	female	female	male		
Experience	expert	novice	novice	expert	novice	expert	novice	expert	novice	novice	novice		

in collaboration between two laboratories from the University of Siegen, Germany, and the University of Colorado Boulder, United States of America. In total 24 subjects participated in the study. Participants from Germany were mostly players from a

semi-professional basketball team that participates actively in a basketball league. Participants from the USA were mostly graduate students with mixed prior experience in basketball. We originally included a void class for miscellaneous movements outside of the primary labeled ones, such as drinking from a water bottle or tying shoes. These were mostly performed during rest breaks. The samples annotated as void resulted in an irrelevant small class, which could not be recognized by our classifier because they are most often performed in conjunction with one of the locomotion classes. We ultimately decided against including this void class, since it was very rare that players were not performing one of the 10 classes of *locomotion* or *basketball* activities. However, the data that is not annotated as one of the aforementioned classes are categorized as *not_labeled*. This class can be seen as a very noisy but realistic void class that can be used by researchers who focus on deeper insights in the NULL-class problem defined by Bulling *et al.* [45] or who would like to evaluate deep learning architectures that are focused on the robust classification of void data. This class mostly contains data during resting periods or transitions between sessions. However, since the data is recorded under real-world conditions, many participants did not sit and rest during these periods, instead, they tended to walk through the gym, shoot the ball, or perform individual dribbling exercises. One of the players, namely *2dd9_na*, wore the smartwatch accidentally on their non-dominant hand, out of habit. We decided to keep this participant in the dataset since this participant represents something that could easily happen in real-world scenarios. Therefore, we think that the participant has added value to the dataset and can be useful for certain studies at a later date.

Preprocessing: We decided to keep the preprocessing on the raw data from the smartwatches to a minimum, as these were already provided with a timestamp and in the g unit. The smartwatch's accelerometer samples' timestamps contained slight ($<$2%) deviations, so we adjusted the time-series by resampling to ensure that all data maintains exact 50 Hz equidistant timestamps. Other common methods of preprocessing inertial data for activity recognition, such as rescaling or normalization to improve machine learning results, were not applied.

Labeling: The data was labeled by two experts, one from each institute, and labeled on 4 different layers: (I) *coarse*, (II) *basketball*, (III) *locomotion*, and (IV) *in/out*. After both experts had finished the labeling, the labels were checked again by expert 1 using visual inspection, see Figure 4.10, and corrected if necessary. Using this labeling methodology, we aimed to obtain as precise as possible annotations with human annotators, where some degree of human error and mislabeling cannot be completely ruled out. Especially in the game phase, activities are often performed both quickly and briefly, which can lead to minor deviations in labels between manual annotations.

Specifically, (I) coarse separates the samples into different sessions, including (1) *warmup*; (2) drills: (a) *sitting*, (b) *standing*, (c) *walking*, (d) *running*, (e) *dribbling*, (f) *penalty_shots*, (g) *two_point_shots*, and (h) *three_point_shots*; (3) *game*; and (4) *in/out*. By keeping the information whether a shot is either a (f) *penalty_shots*, (g)

two_point_shots, or (h) *three_point_shots* later studies can use these labels to distinguish between different shot distances. The label (3) *game* indicates when a game was played. The study from Germany contains 2 game sessions with ~10 min each and the study conducted in the USA contains one session of ~22 min. The two layers (II) *basketball* and (III) *locomotion* contain the labels that correspond to one of the classes shown in Table 4.5, as well as the label *not_labeled*, which is used whenever the information of what exactly a player is doing at a specific moment, could not be seen in the ground truth video or between sessions. The fourth layer in/out is only relevant during the game session since this layer indicates whether a player is on the court or not. However, this layer can be seen as additional meta-information, which can be relevant for future researchers. It has not been used for deep learning validation, since the challenge of classifying if someone is active or non-active seems to be trivial in this scenario.

Class Definitions: The following Table 4.5 contains the class descriptions and Figure 4.11 visualizes one example for each class. Sitting and standing mostly show

Table 4.5 Detailed class description for every class included in the dataset. The dataset is multi-tier labeled with 4 different layers (I) Coarse, (II) Locomotion, (III) Basketball, and (IV) In/Out. The coarse layer is not listed, since it is meant to indicate to which session an activity belongs. Relevant classes are classes 2–13. However, the classes *in* and *out* were not used in our validation.

All Layers			
1. not_labeled	All samples in between sessions, or if it was not possible to recognize the activity in the video (e.g. due to occlusions).		
In/Out			
2. In	Indicates that the subject is currently actively participating in the game.	**3. Out**	Indicates that the subject is currently not actively participating in the game. This class mostly included sitting or walking.
Locomotion		**Basketball**	
4. sitting	Sitting on the floor or the reserve bench.	**9. dribbling**	Dribbling while performing one of the following locomotion activities: (3) standing, (4) walking, (5) running.
5. standing	Standing still.	**10. shot**	A basketball shot with and without a jump. Included are penalty shots, 2-point and 3-point shots.
6. walking	Walking at the average walking speed of a human (4-5 km/h).	**11. layup**	A layup is a complex class that involves: grabbing the ball, making 2 steps (FIBA) or 3 steps (NBA), jumping and putting the ball in the basket.
7. running	Running is a metaclass for all velocities of running. Therefore, it contains jogging (5-6 km/h), fast running (6km/h <10 km/h) and sprinting (>10km/h).	**12. pass**	Passing the ball. Included are chest passes, bounce passes, overhead passes, one-handed push passes and so-called baseball passes.
8. jumping	A jump typically is part of a more complex activity, like (10), (11) or (13).	**13. rebound**	The player jumps and catches the ball mid-air with one or two hands.

sedentary acceleration patterns with sporadic movements of people moving their wrists, while walking and running show the commonly known oscillating patterns. Dribbling can vary depending on how a player is dribbling. For example, a player can dribble with their dominant or non-dominant hand, dribble the ball by switching hands, or do even fakes and tricks. These styles have slightly different characteristics and can be distinguished, see [89]. However, we decided to summarize these differences in one class. Even when the ball is dribbled with the non-dominant hand, the data from the dominant hand shows the oscillating characteristics of the dribbling movement. Jumping is an assembled class that also includes jumps belonging to either a shot, rebound, or layup activities. These classes share the trait that the jump - a peak on the coronal plane - is clearly visible. However, the classes differ mainly in the sensor

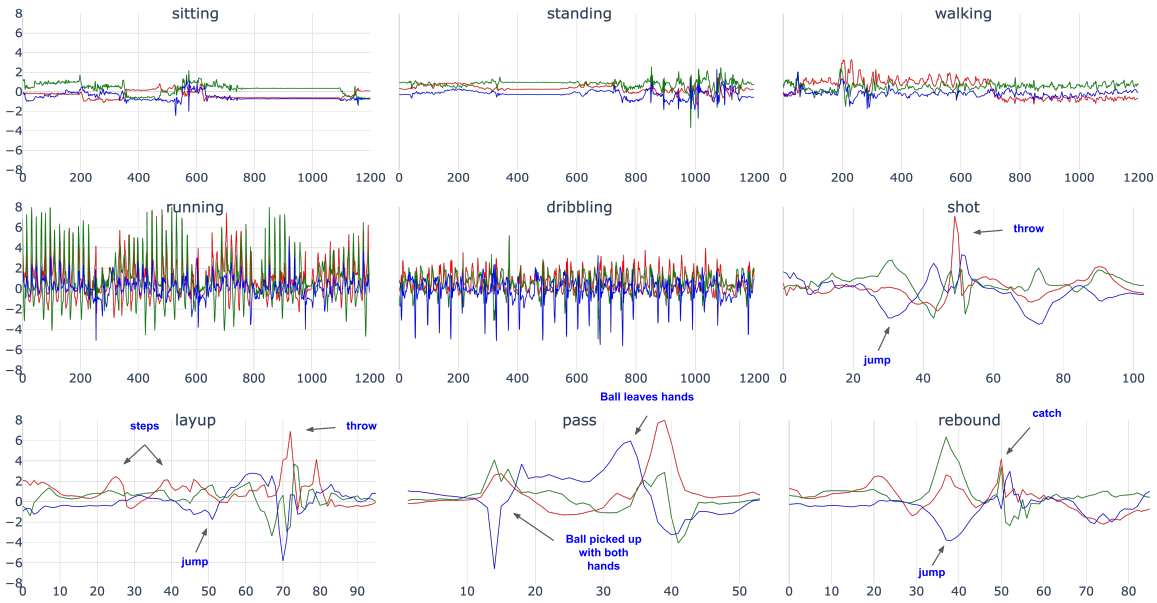
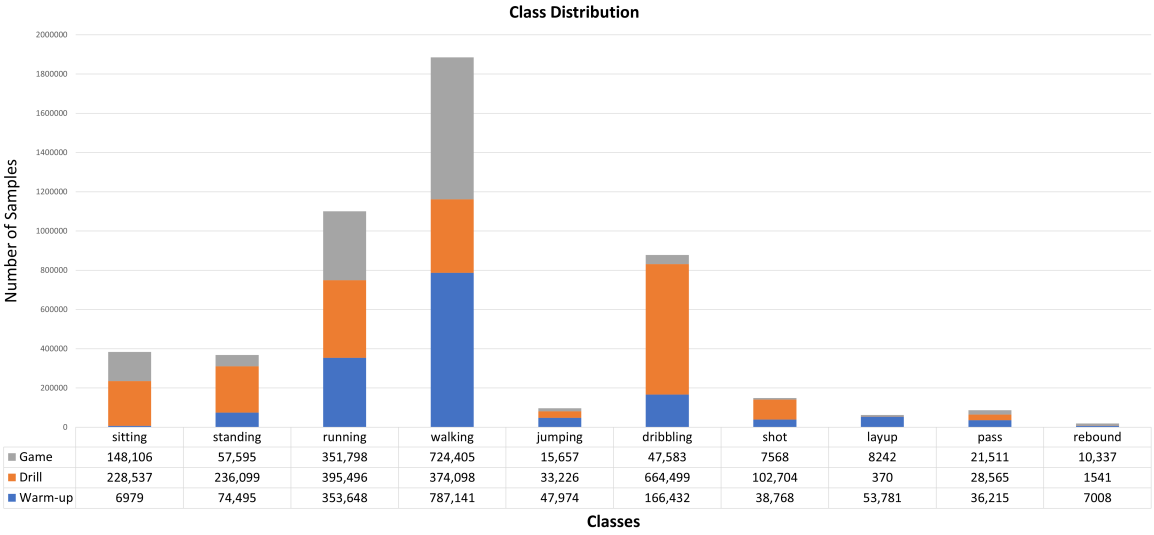


Figure 4.11 Exemplar time-series data for the included activities. The examples shown for the periodic activities *sitting*, *standing*, *walking*, *running*, and *dribbling* contain 1200 samples (approx. 24 s). In order to better represent the complex activities *shot* and *layup* as well as the micro-activities *pass* and *rebound*. Jumps are marked in classes where the activity occurs. Such short periods were summarized in the activity *jumping*.

data prior to the peak. The shot contains the player grabbing and lifting the ball before jumping mostly straight up to shoot or, in the case of a penalty shot, performed in a standing position. A rebound is mainly a clear jump upwards or in the forward direction and a layup contains the combination of running 2 or 3 steps (depending on FIBA or NBA rules), a jump, and throwing the ball in the basket while jumping forward. The pass is a very short activity characterized by a forward acceleration on the sagittal plane. Figure 4.12 shows how the classes are distributed over the sessions.

As one can see, locomotion activities such as *walking* and *running* are distributed almost equally over the sessions. *Sitting* was not performed during the warm-up session and *layup* is almost exclusively performed during warm-up and the game. Samples labeled as *dribbling* and *shot* were mostly recorded during the drill sessions and, similar to layups, performed way less in the game. *Rebound* is the least recorded activity. The imbalance is caused by the realistic recording setup of the dataset and reflects the reality of a training session including training games in basketball. The imbalance should be considered a challenge rather than an obstacle since all data recorded in real environments show such characteristics. Most of the datasets mentioned in Table 4.2 share an imbalance either with regards to the class distribution or study participant homogeneity. Further, we believe that future studies would benefit from (rather than negatively impacted by) class imbalance and intersubject variability in the dataset. Even though evaluation metrics may not reach their maximum easily, we argue that



	sitting	standing	running	walking	jumping	dribbling	shot	layup	pass	rebound
Game	148,106	57,595	351,798	724,405	15,657	47,583	7568	8242	21,511	10,337
Drill	228,537	236,099	395,496	374,098	33,226	664,499	102,704	370	28,565	1541
Warm-up	6979	74,495	353,648	787,141	47,974	166,432	38,768	53,781	36,215	7008

Figure 4.12 Class distribution of the Hang-Time HAR dataset. The total number of samples per class is: *sitting* : 383,622 (~2.1 h), *standing*: 368,189 (~2.0 h), *walking*: 1,885,644 (~10.5 h), *running*: 1,100,942 (~6.1 h), *jumping*: 96,857 (~0.53 h), *dribbling*: 878,514 (~4,8 h), *shot*: 149,040 (~0.82 h), *layup*: 62,393 (~0.34 h), *pass*: 86,291 (~0.47 h), and *rebound*: 18,886 (~0.10 h). In total: 5,030,378 labeled samples or ~27.7 h of data.

this setup is more realistic and more representative of a recreational sport itself and will help researchers understand open research questions better than a fully balanced dataset.

Combining Classes: The layers provided in our dataset make it possible to extend it with additional and more challenging classes. For example, shots can be distinguished between *penalty_shots*, *two_point_shots*, and *three_point_shots* by taking into account the *coarse* layer. The *locomotion* layer holds the information if the activity *dribbling* was performed while the player was *standing*, *walking*, or *running*. Therefore, the class definitions in Table 4.5 only contain the basic classes and can be extended individually - depending on the requirements of one's project.

4.2.4. Analysis

This section will provide a preliminary inspection of our dataset. The range of methods employed here includes descriptive statistics, baseline statistical analyses, and machine learning performance results. Our feature analysis focuses on experts vs. novices since we believe that this feature is a strong asset of our dataset that needs to be highlighted. Differences in the data with regard to the players' experience are visible through features and can be used in later research to develop systems that react to these differences, such as supporting and accompanying a player in the further development of his/her playing skills. If researchers would like to use the dataset as a benchmark dataset for deep learning experiences, they can exclude one or the other group.

Feature Analysis. Our analysis focuses on the representation of intraclass variability and interclass similarity, as well as clarifying the differences between novices and experts. Figure 4.13 contains the raw data of approximately 7 min of dribbling while the person stands, walks, and runs with different velocities. The locomotion speed increases over time. Through visual inspection, we can already clearly see that the dribbling patterns differ greatly between novices and experts.

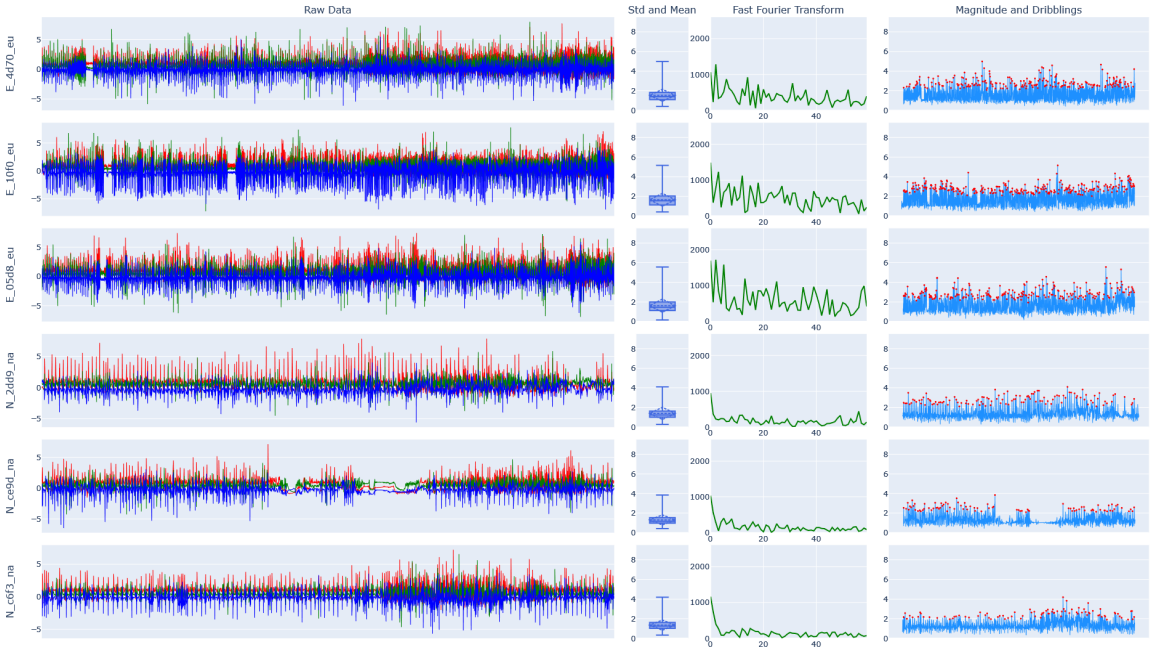


Figure 4.13 Feature analysis of the class dribbling for players 4d70_eu, 10f0_eu, and 05d8_eu (experts) and 2dd9_na, ce9d_na, and c6f3_na (novices). The plot consists of 4 columns. (1) Raw data as recorded during the dedicated dribbling drill (approx. 7 min of data (Germany) and 5 min of data (USA). The X-axis is represented in red, the Y-axis in green, and the Z-axis in blue color. (2) Standard deviation (diamond shape), median, interquartile q1 and q3 (rectangle shape) as well as upper and lower fences. (3) Fast Four Transform [203]. (4) Local maxima [204] (*prominence = 1.4*) calculated using the magnitude of the input signal (1), every red dot indicates a peak that is interpreted as one dribbling.

The results of the Fast Fourier Transform (FFT), visible in column (3), indicate that expert players dribble the ball with a wider frequency spectrum than novices, caused by variations in the dribbling style (changing hands, dribbling low/high or fast/slow, or doing tricks). Furthermore, the expert players show a higher mean frequency as well as a higher magnitude column (4) than novice players.

However, this is explainable since player *b512_na* mostly dribbled the ball at a walking pace (visible in the video footage) and player *0846_eu* has intermediate dribbling skills, even though the overall skill level can be categorized as a novice. Additionally, for the features depicted in Figure 4.13, we calculated the arithmetic mean of dribble per second AM.D. and the Signal-to-Noise-Ratio (SNR), as defined in

the following equation.

$$SNR_{db} = 10 \cdot log_{10}\left(\frac{Psignal}{Pnoise}\right) \tag{4.1}$$

The experts have a higher rate of dribbles per second than the novices, and this shows that experts dribble the ball more comfortably resulting in fewer ball losses and a faster pace, as shown in column (4) of Figure 4.13. More significance is illustrated in the SNR between the two groups. A higher value means that more noise is present in the signal, or the ball is dribbled in a less controlled manner, and this is also visible in the raw data of Figure 4.13.

The Principle Component Analysis (PCA) [184], shown in Figure 4.14, calculated for the same participants shows, exemplary on the basis of the classes *shot* and *layup*, the intraclass variability but also the interclass similarity mentioned at the beginning. The first column contains all subjects, the following three columns contain the experts, and the last three columns contain the novices. The PCA shows that novices follow less coherent movement patterns. Experts, however, present more similar patterns, which differ minimally on both component axes of the PCA.

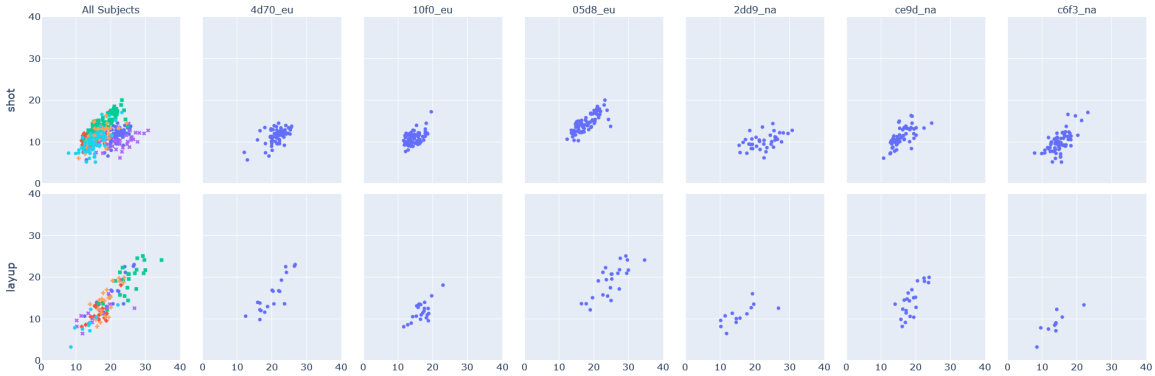


Figure 4.14 Principle Component Analysis of the classes (1) shot and (2) layup. For the same subjects mentioned in Figure 4.13 and Table 4.6. The colors represent the 6 different participants included in this figure. *4d70_eu* is represented in blue, *10f0_eu* in red, *05d8_eu* in green, *2dd9_na* in purple, *ce9d_na* in orange, and *c6f3_na* in turquoise.

Table 4.6 Arithmetic Mean of dribbles/second (AM D.) and Signal-to-Noise-Ratio (SNR) are listed per subject and separated between *experts* and *novices*.

	Experts			Novices		
ID	**10f0_eu**	**05d8_eu**	**4d70_eu**	**2dd_na**	**c6f3_na**	**ce9d_na**
AM D.	1.10	1.05	1.04	1.01	1.02	1.01
SNR	3.40	2.97	3.47	5.93	8.43	7.17

The following Figures 4.15 and 4.16 show 10 examples for the same six participants used in the figures before. The shots show participant-independent patterns that include a negative peak on the z-axis (jump) followed by a positive peak on the y- and z-axis (shot). Such a coherent pattern is hardly visible for the *layup* class.

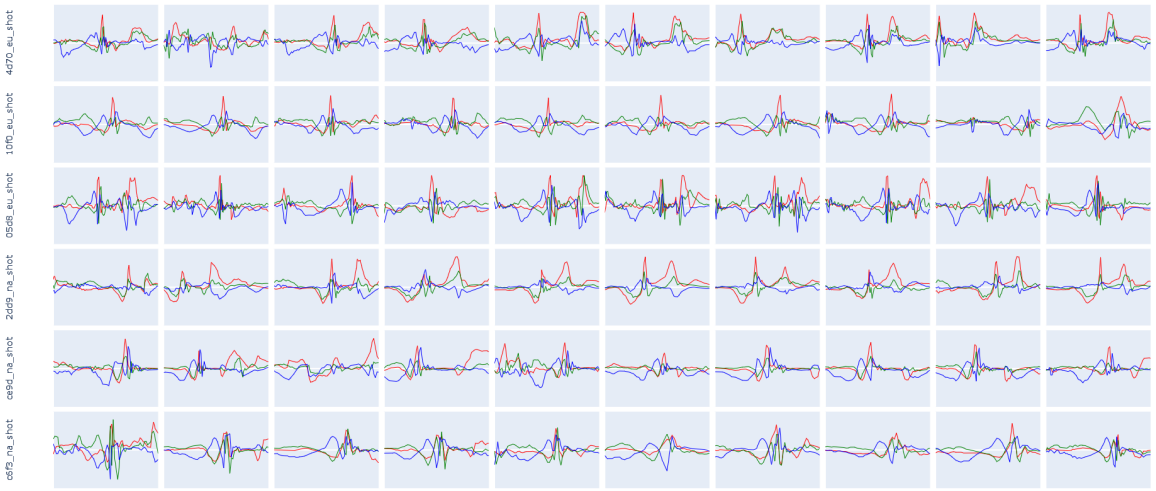


Figure 4.15 Ten instances of the class *shot* for the same subjects as mentioned in Figures 4.13 and 4.14. A clearly visible pattern can be seen in all examples. The length of the activity typically varies between 1000 and 3000 ms with an average duration of approx. 1700 ms, depending on the subject and the execution-style. The X-axis is represented in red, the Y-axis in green, and the Z-axis in blue color.

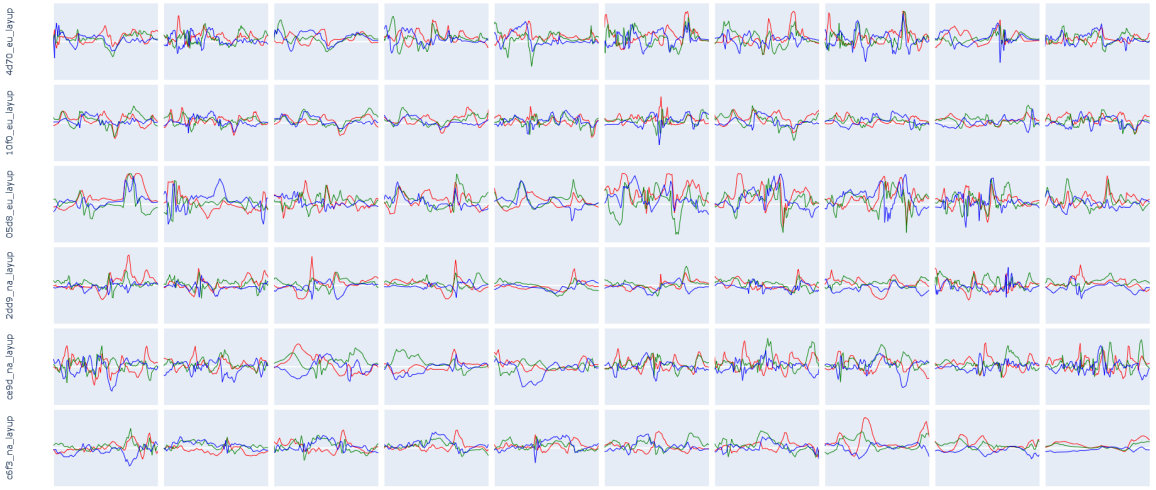


Figure 4.16 Ten instances of the class *layup* for the same subjects as mentioned in Figures 4.13 and 4.14. The patterns vary by subject and sometimes even differ between instances of the same subject. This class inherits a strong intraclass and intersubject variability. The length of the activity typically varies between 1000 and 3000 ms, with an average duration of approx. 2000 ms, depending on the subject and the execution-style. The X-axis is represented in red, the Y-axis in green, and the Z-axis in blue color.

Such perfect examples, as seen in Figure 4.11 are rare for the *layup*, especially when the player is contested. The intensity, as well as the execution of the activity, differs a lot depending on the situation.

Deep Learning Analysis. We investigated a variety of prediction scenarios to provide a first impression of potential test cases and benchmark scores that can be achieved using the Hang-Time HAR dataset. As architectures for our classifiers, we chose to use both a shallow variant of the DeepConvLSTM network [35] and Attend-and-Discriminate network [3]. Each of the defined training scenarios employs either a Leave-One-Subject-Out (LOSO) cross-validation or a train-test split to evaluate a network's predictive performance. The former (LOSO) involves each subject becoming the validation set once while all other subjects are used for training the network, while the latter (split), as the name suggests, simply splits the data into two parts - one used solely for training and the other used solely for testing. During all experiments, we employ a similar hyperparameter setup as used in [35]. We further alter the architecture suggested by Abedin *et al.* [3] to encompass the findings discussed in [35], i.e. employing a 1-layered recurrent part and utilizing 1024 hidden recurrent units for both architectures. Lastly, in order to minimize the effect of statistical variance, for each test case we calculate the average predictive performance across 3 runs, each time employing a different random seed drawn from a predefined set of 3 random seeds. In order to determine a suitable sliding window size, three different window lengths, i.e. 0.5, 1, and 2 seconds, with an overlap of 50% were evaluated using a LOSO cross-validation on the complete Hang-Time HAR dataset. Amongst the tested window lengths results showed little to no difference with the standard deviation of the macro F1-score ranging only between 0.4% (shallow DeepConvLSTM [35]) and 1.2% (Attend-and-Discriminate [3]). Nevertheless, we determine a sliding window length of 1 second with an overlap ratio of 50% to be most suited for the Hang-Time HAR dataset as we expect that:

(a) A smaller window length would not be able to capture enough data, and thus patterns specific to activities, which could be learned by the network.

(b) A larger window length would capture too much data, increasing the risk of patterns specific to short-lasting activities being mixed with patterns of other activities. This would make it less likely that a network learns to attribute only relevant patterns to short-lasting activities.

During our experiments, we are investigating how well our network generalizes in two regards:

(a) *Subject-independent generalization:* As with almost any activity, basketball players tend to have their own specific traits in performing each basketball-related activity. Within these test cases, we investigate how well our network generalizes across subjects by performing a LOSO cross-validation on the drill and warm-up data of all subjects. During each validation step, the activities of a previously unseen subject are predicted, and thus the experiments will determine how well our network generalizes across subjects and whether subject-

independent patterns can be learned by our architecture.

(b) *Session-independent generalization:* As previously mentioned, data recorded during an actual basketball game can heavily differ from "artificial" data recorded during the drill and warm-up sessions, as subjects did not have to adhere to any (experimental) protocol. Thus, the session-independent test cases investigate how well our network predicts the same activities performed by already-seen subjects during an actual game. Within these experiments, we train our network using data recorded by all subjects during the drill and warm-up sessions and try to predict the game data of said subjects. These type of experiments will give a sense of how well our network is able to generalize specifically to real-world data and simulates the transition from a controlled to an uncontrolled environment. The network learns player-specific patterns from the warm-up and drill sessions and tries to classify the more dynamic game subset.

In the following, the results obtained during the two test case types will be illustrated. All results as well as the raw log files of each test case can be found on the projects' Netpune.ai page[2]. The used architectures and the code of the performed experiments are published in our GitHub repository[3]. Looking at the results in Figure 4.17 and 4.18 one can see that there are major differences regarding how well our network generalized across study sessions (parts) and across subjects. Overall one can see that

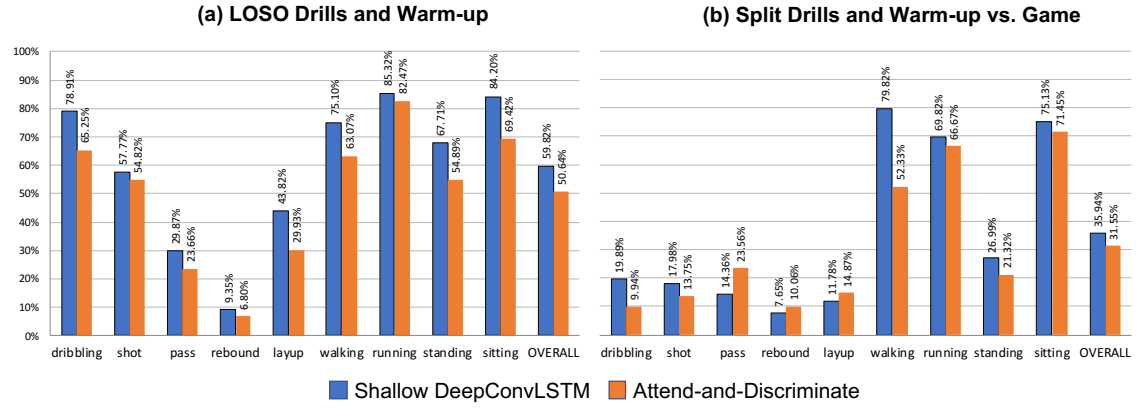


Figure 4.17 Overall results of the deep learning experiments using a shallow DeepConvLSTM [35] (blue) and Attend-and-Discriminate architecture (orange). Both models were trained with a 1-layered recurrent part with 1024 hidden units and a sliding window of 1 second with 50% overlap. The left plot (a) shows the per-class LOSO results obtained from training on the drill and warm-up data. The right plot (b) shows the per-class results predicting the game data when trained on the drill and warm-up data. All results are averages across 3 runs using a set of 3 random seeds. Both architectures suffer a significant loss in predictive performance when being applied to in-game data, i.e. data recorded in an uncontrolled environment.

using the Hang-Time HAR dataset as input both architectures did not generalize well

[2] `https://app.neptune.ai/o/wasedo/org/hangtime`
[3] `https://github.com/ahoelzemann/hangtime_har`

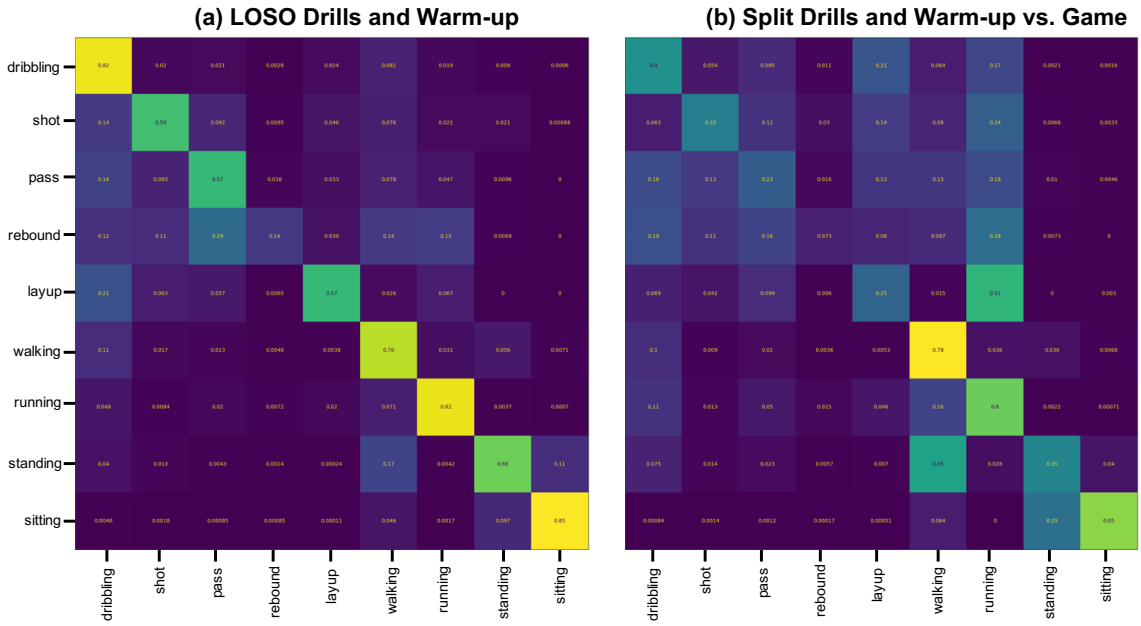


Figure 4.18 Confusion matrices of a shallow DeepConvLSTM [35] applied to the Hang-Time dataset. The model was trained with a 1-layered recurrent part with 1024 hidden units, a sliding window of 1 second with 50% overlap, and a fixed random seed. The left confusion matrix (a) is obtained from averaging the per-subject LOSO results using the drill and warm-up data as input data. The right confusion matrix (b) is obtained from training on the drill and warm-up data and validating on the game data. One can see an increase in overall confusion when applying the architecture to in-game data, i.e. data recorded in an uncontrolled environment.

across sessions, i.e., from drills to games. Looking at the subject-independent results one can see that almost all classes tend to transfer well with only layups ($< 45\%$ macro F1-score), passing ($< 30\%$ macro F1-score), and rebounds ($< 10\%$ macro F1-score) as outlying activities, with the average macro F1-score above 50% for both architectures. Contrarily, the session-independent results show a significant decrease in overall predictive performance by around 24% for the shallow DeepConvLSTM [35] and around 19% for the Attend-and-Discriminate [3] architecture. Nevertheless, this trend does not apply to all activities equally, with most locomotion activities (walking, running, and sitting) not as heavily affected ($< 50\%$ macro F1-score) in prediction performance as the basketball activities (dribbling, shooting, passing, rebound and layup) whose macro F1-scores do not exceed 20% for both architectures. We accredit this drop in performance to the fact that basketball games by nature have more unforeseen situations to which players need to adjust their movement too. In general, it is rare that players are able to perform e.g. an uncontested layup (e.g., certain fastbreak situations) resulting in altered feet and arm movement in order to find the necessary space and successfully score. The influence of a game-like situation can particularly be seen in the locomotion activity standing which sees a major decrease when trying to be predicted in-game. Players constantly move to defend an

oncoming player of the opposing team, which makes standing in-game very different from standing during drills, as players are e.g. going into a defensive position or are keeping in contact with their assigned player on defense.

We identify the challenges for future research and experiments to be two-fold:

(a) Results obtained during session-independent experiments show the poor generalization of basketball-related activities from controlled to uncontrolled environments. This further underlines the bias introduced by researchers when relying on data recorded in a controlled environment compared to uncontrolled environments. It is to be investigated whether it is possible to increase generalization through means of altering the training process or employing architectures.

(b) Employing the definition as defined in [34], Hang-Time HAR offers both complex (shot, layup) and sporadic (rebound, pass) activities. As said activities are not as reliably detected (even in controlled an environment) as other activities, it is to be investigated whether this lies in the nature of the activities, or can be accredited to the employed network architecture reaching its limits.

4.2.5. Discussion

We present our dataset Hang-Time HAR, an extensive dataset for (Basketball) Activity Recognition. The dataset was recorded in two different sessions and continents (using two different sport-specific rule sets) in a real-world scenario with approximately 2266 min of real basketball training sessions and training games. The dataset we introduce offers a large variety of activities performed by 24 subjects in both (partly) scripted (drill and warm-up) and unscripted (game) recording sessions. Activities range from simple ones, such as a player‘s locomotion, to complex ones, such as layups and shooting which consist of in-activity sequences. Each basketball player was equipped with a single wrist-worn inertial sensing smartwatch, and labeling was performed by annotating video footage of the sessions.

The feature analysis shows that Hang-Time HAR has considerable intraclass variability and interclass similarity as described by Bulling *et al.* [45]. This effect was strengthened by the recording setup of a semi-controlled environment. From the perspective of deep learning for human activity recognition, the dataset offers a variety of new challenges. As evident in the results of our Deep Learning analysis, during the LOSO cross-validation the architectures we chose reached their limits with respect to the classes *rebound* and *layup* in both session types we evaluated, see Figures 4.17 and 4.18. To be able to recognize it in a LOSO cross-validation, where no prior information on the subject is given to the classifier, we need either more samples of that class, e.g., through applying techniques such as data augmentation or a deep learning architecture that is able to handle under-represented classes. Basketball-specific classes were predicted during the game, on average, with a 25% F1-Score. *Passes* and *rebounds* were extremely difficult for the classifier to detect since their execution time is often under 1 s. Furthermore, the most significant part of the activity *rebound* is the jump - which is a sub-activity that is shared with other classes such as *shot* or *layup*. These activities

that were predicted poorly when missing subject-specific training data also correspond to the fewest samples in the dataset. Future work could involve testing techniques such as artificially increasing the under-represented classes through data augmentation or a more suitable deep learning architecture for handling class imbalances. According to Bock *et al.* [34] we distinguish between sporadic, simple/periodical, transitional, and complex activities. However, datasets shown in Table 4.2 mostly focus on locomotion activities and activities of daily living. Only a few, such as [177, 55, 36, 122], include sporadic, transition or complex activities, and many datasets that do include sports [11, 20] aggregate an entire sport into a single activity. Published sports studies tend to not release their datasets publicly or only upon request - with Trost *et al.* [208] and Bock *et al.* [36] as the only exceptions, as shown in Tables 2.2 and 2.3. As a result, sports-specific IMU-based datasets available to the public that reflect the complexity and characteristics of a specific sport are very limited. Due to the nature of the sport of basketball, our dataset contains classes where the characteristics mentioned by Bock *et al.* apply. *Rebound, pass, and jump* can be considered as sporadic classes. The locomotion classes - *sitting, standing, walking*, and *running* are periodic classes, *shot* and *layup* contain complex, interrelated activities. During the warm-up and game, all activities were situation-based and therefore the dataset contains natural and fluently performed transitions between classes as well as overlapping activities.

By using the different semantic layers of the dataset - *coarse*, *basketball*, *locomotion*, and *in/out* - researchers are able to focus on different aspects of activity recognition by studying the different semantic levels either in isolation or in combination and incorporate them in their research appropriately. In particular, the combination of various semantic levels offers researchers the possibility to design and develop game analysis algorithms based on IMU signals. Such algorithms could either analyze the players' performance with or without focusing on specific activities or could analyze the game itself in a holistic approach. Wrist-worn smartwatches - as used in this study - are not allowed to be worn during an official basketball game, since they bear the risk of harming the player or other players on the field. However, we believe that it can be replaced in future studies, e.g., by sweatbands that incorporate IMU sensors. Such a device could come in a similar form as those in [162] or [19].

Apart from providing a benchmark dataset for future machine and deep learning studies, we believe that our dataset has cross-domain application purposes that can help to solve open research questions such as activity recognition of complex classes in real-world scenarios, including the development of preprocessing or postprocessing algorithms for real-world data as well as designing neural network architectures for such scenarios. In particular, the game data introduces a completely new scenario for human activity recognition in which activities overlap each other and are performed with a higher pace and altering patterns, due to the ball possession and the psychological pressure during a game situation. Such semantic learning can become another important sub-field for HAR in the near future, as demonstrated by the recently published architecture SemNet by Venkatachalam *et al.* [218].

It is known that transfer learning for HAR does not perform as well as it does for

vision data. Pretraining and transferring a neural network do have a positive effect on the classifiers' capabilities as well as the training time [95]. Our dataset can help explain these phenomena since the locomotion layer is class-wise compatible with many other datasets shown in Table 4.2 and should be therefore transferable. However, due to the recording environment and activity domain, the classes are expected to differ from similar classes published by datasets shown in this table. Further transfer learning studies that test the effectiveness of pretraining a neural network model with regard to several domain-specific datasets can be of interest to the activity recognition community. We think that pretraining a neural network on a sports dataset and transferring the model to another sport can have a higher positive impact on the classifier than pretraining it on a non-domain dataset. However, this is speculative at this point and needs to be investigated by future studies. The different skill levels of our participants shall not be seen as a disadvantage, but rather as a unique feature that opens up challenges and opportunities for not yet addressed research questions. The distinction between the two levels of skills can help understand the real effect of noisy real-world performed instances of activities on a trained classifier. As Section 4.2.4 describes, we were able to identify differences in the patterns between *novices* and *experts* due to unclean performed activities, such as shooting the ball with both hands or dribbling the ball with less control than experienced players. Including or excluding one or the other will have an effect on the classifier. We think that these effects are valuable and should be investigated as part of a larger and more complex study in the context of classifier poisoning, transfer learning, or research problems with regard to data labeling.

An IMU-based approach has the advantage over vision-based approaches since wearables (e.g., smart watches) are low-cost, widely available, and quickly deployable to players on every court (indoor and outdoor). Vision-based approaches require a more complex tech build, which is cost- and labor-intensive to set up and configure. Furthermore, in future works, a simple model can be trained and deployed on a wearable device in order to classify motions through IMU data in real-time and on-device. Recent advances, such as the TinyHAR [243], are capable of detecting human activities with fewer trainable parameters and are therefore power-efficient enough to be deployable on wearable devices, such as the Bangle.js smartwatch (the Bangle.js comes with TensorFlow Lite preinstalled on the microcontroller). We, therefore, expect that our benchmark dataset will have a significant impact on activity recognition research in itself, but also encourage more follow-up work in the methodologies for designing, recording, and annotating such datasets. We argue that the sports domain, in general, offers researchers a recording environment that can range from a controlled to an uncontrolled setting with the advantage that data can be labeled retrospectively using video footage.

The players' meta information can be used to gain deeper insights into how a person's build and sports experience affects the execution style of an activity. Even though the metadata contains basic information about the players' prior basketball experience, it does not claim to evaluate the playing skills of individual players. The video footage

might be used to provide such annotations to be added as extra annotation layers. In future work, we would like to add another layer to the game sessions called def/off, which indicates whether a player is currently playing defense or offense, respectively. This information is useful with regard to a player's locomotion since the defense position is usually played in an upright position with hands raised and knees slightly bent, see Figure 4.7. Furthermore, since this paper focuses on feature analysis and deep learning-driven classification methods, medical statistics such as a Bland–Altman analysis [29] and biomedical derived studies [22, 114, 46], fall out of our scope of study but could supplement this paper in future works.

Besides the use case of basketball activity recognition, we expect that certain activities are generalizable across different sports. Periods in which a player did run without dribbling (the periods can be filtered by taking into account the different semantic levels of annotations) can be transferred to sports such as handball, indoor soccer, futsal, or in general indoor sports that share similar field size and have periods of players running without a ball. The dribbling movement seems to be transferable between basketball and handball. However, we expect that the transferability will have its limitations. For example, the class jumping will have different characteristics in volleyball compared with a jump in basketball, since the game volleyball itself is more focused on the vertical space and has different patterns depending on what action the players perform. Volleyball has basically six different skills that players perform during a game, which are: serve, pass, set, attack, block, and dig. All of them, except two special variants of serve and pass (float serve and forearm pass) involve jumping. Hypothetically, if a wrist-worn sensor-based dataset with volleyball activities were published, it would be interesting to explore whether the class jumping would be transferable.

4.2.6. Conclusions

During this study, we have developed a basketball activity dataset that brings a variety of unique features with it and is, to the best of our knowledge, the only sensor-based and publicly available activity recognition dataset that focuses on fine-grained team sports activities. The dataset introduces data from wrist-worn inertial sensors of 24 players from two teams and recorded in two different continents where slightly different rule sets are applied. The participants perform ten different basketball activities that are grouped into four different semantic levels. The dataset contains warm-ups, drills, and game phases. Typical routines were followed during the drills but not during the warm-up and game, where players were allowed to play as they preferred. Therefore, this dataset contains data from controlled as well as uncontrolled environments which can be filtered as needed by researchers. The different semantic levels of the annotations make it not only possible to focus on general locomotion or specific basketball activities, but also to create more complex classes as mentioned before. The two levels of skills, novice, and expert, inherit a strong intraclass and intersubject signal-variability which has already been mentioned by Bulling *et al.* [45] in 2014 and is still an ongoing research challenge in real-world scenarios. Therefore,

we argue that this feature is directly relevant to real-world activities of any domain and can be used to investigate these problems further. As aforementioned, the class *not_labeled* contains data that corresponds to NULL activities as well as activities that are not part of the dataset. As such, this class represents a very realistic and naturally-designed *void* class which can be of interest to studies that focus on investigating the NULL-class problem. The results of our deep learning analysis show that current architectures are not capable of detecting complex classes. In order to overcome this obstacle, further research on data preprocessing and architectural neural network design is needed. This problem becomes more challenging if the data is recorded in a real-world and uncontrolled environment.

Given its uniqueness as a fine-grained sports dataset, class variability, high number of study participants, and comprehensive coverage of rule-set-varying basketball characteristics, we firmly believe this dataset will suit evaluating machine learning and deep learning algorithms, network architectures, and previously mentioned problems. The dataset should also serve as an ideal benchmark for the human activity recognition community across application domains.

Section 4.3

Summary

While initially a straightforward attempt to expand activity recognition to a specific sport, like basketball, the initial feasibility study [89] sparked the inception of this thesis and led toward further exploration of concepts interconnected with activity recognition in the domain of sports and beyond. The feasibility study addressed recognizing diverse basketball activities including *shooting* and *dribbling*, as well as discerning between three distinct dribbling methods: *low*, *medium*, and *high dribbling*, in addition to the execution of a *crossover* maneuver. It furthermore demonstrated that even traditional machine learning techniques, such as Random Forest and k-nearest-neighbor, have the capacity to effectively classify complex basketball activity patterns using data obtained from wrist-worn sensors. We, therefore concluded that more sophisticated deep learning techniques could potentially detect additional nuanced activity classes.

Based on these findings, the concept emerged to develop a comprehensive dataset that not only portrays basketball activities but also captures group behavior during basketball game sessions. However, this idea was deferred until the finalization of the Activate System, see Section 3.1, to ensure sufficient simultaneous equipping of subjects with wearable sensors, and further postponed until COVID-19 pandemic-related restrictions were lifted. Before engaging in dataset development and its subsequent utilization, I hypothesized that trained neural networks, such as the Deep Convolutional LSTM [159] or Attend-And-Discriminate [3], could effectively distinguish various basketball classes. Through assessment of the conducted tests and feature analysis, I conclude that the multi-class problem was successfully classified with high F1-Scores for certain activities. Overall, I demonstrate that activity recognition can effectively work on a granular scale for specific sports when sensors are strategically placed at biomechanically relevant body locations. However, further investigation is requisite to accurately classify a dataset encompassing periodic, complex, and spontaneous actions. These classes in particular posed challenges for the trained models.

We hold the perspective that the dataset we've presented holds potential value for fellow researchers focusing on advancing exploration into group activity recognition or detecting complex and spontaneous classes as described in our publication. The dataset is further strengthened by the inclusion of an uncontrolled game session, which closely replicates a real-world scenario. This uncontrolled environment provides valuable insights into sensor behavior in more natural conditions. Participants engaged naturally, simulating a regular training session, thereby ensuring unbiased sensor data and mitigating any potential Hawthorne Effect [90]. Therefore, Hang-Time HAR exhibits characteristics of both controlled, protocol-driven datasets with fixed sessions, as well as uncontrolled real-world datasets with strong participant and situation-dependent activity patterns. This attribute is pivotal for subsequent investigations.

Chapter 5

Deep Learning for Human Activity Recognition

The advent of deep learning in the last decade is followed by a shift away from traditional machine learning, resulting in remarkable improvements in capabilities across all domains of recognition or projection problems. A key advantage of deep learning is the ability of deep neural networks to automatically learn feature representations from raw input data, bypassing manual feature engineering that relies heavily on domain expertise. This data-driven feature learning improves efficiency while requiring minimal human effort and domain knowledge.

This chapter focuses on a critical investigation of specific deep learning techniques, including transfer learning and data augmentation, and their application to HAR data plus limitations in this domain. The research questions framing my exploration are:

(a) **Transfer-Learning Across Datasets or Sensor Positions: Feasibility and Benefits**
Can model transfer between diverse datasets or sensor positions effectively enhance classification outcomes? I assess the viability of cross-dataset and cross-sensor transfer learning, examining its potential advantages and limitations.

(b) **Assessing the Impact of Sensor Orientation Alignment on Classification**
How does aligning sensor orientations, for instance through axis inversion, influence classification accuracy? I'm exploring the impact of orientation alignment on model performance. By aligning the data points in a specific orientation, I aim to determine if this improves the effectiveness of the model.

(c) **Comparing Data Augmentation Sample Selection Strategies**
Data augmentation is a prominent strategy to expand training datasets and mitigate overfitting. I evaluate the significance of employing participant-wise and fold-wise selection methods for data augmentation, shedding light on their respective effectiveness.

Section 5.1

Transfer Learning for Human Activity Recognition

[95] Hoelzemann, Alexander, and Van Laerhoven, Kristof
Digging deeper: Towards a Better Understanding of Transfer Learning for Human Activity Recognition
Proceedings of the 2020 ACM International Symposium on Wearable Computers, 50-54
https://doi.org/10.1145/3410531.3414311

Portions of the original publication have been removed or edited for inclusion in this thesis. However, no changes were made that altered the results or conclusions presented in the original work.

Contributions:

- Both authors designed the study.
- I implemented the study and analyzed the results.
- Kristof Van Laerhoven guided this work, assisted in the methodologies, and helped with analyzing the results.

Transfer Learning is becoming increasingly important to the Human Activity Recognition community, as it enables algorithms to reuse what has already been learned from models. It promises shortened training times and increased classification results for new datasets and activity classes. However, the question of what exactly is transferred is not dealt with in detail in many of the recent publications, and it is furthermore often difficult to reproduce the presented results. Therefore we would like to contribute with this paper to the understanding of transfer learning for sensor-based human activity recognition. Our experiment uses weight transfer to transfer models between two datasets, as well as between sensors from the same dataset. As source- and target-datasets PAMAP2 and Skoda Mini Checkpoint are used. The utilized network architecture is based on a DeepConvLSTM. The result of our investigation shows that transfer learning has to be considered in a very differentiated way since the desired positive effects of applying the method depend very much on the data and also on the architecture used.

5.1.1. Introduction

The recording of datasets is always associated with a very large investment of time and energy and is also always accompanied by significant costs. For this reason, the community has relatively few datasets at its disposal that are suitable for training neural networks due to their nature, with respect to scope, quality, and reliability. Therefore, algorithms have been increasingly the focus of research in recent years, which

either allow to enrich datasets with information at low cost or to reuse information from already learned models, like Transfer Learning.

Transfer Learning is a Machine Learning technique with which we are able to transfer knowledge from one previously-trained model to another and therefore use this knowledge to solve a similar problem. Many of the already published papers in which Transfer Learning is used for Human Activity Recognition focus mainly on the feasibility of the methodology or on improving the classification results on the target dataset by adapting the used network architecture. As a result, Transfer Learning for Human Activity Recognition still contains many unknown aspects. However, since this technique has great potential to improve classification results and to reduce the computational time for training neural networks, it is necessary to do more research on this topic and put the spotlight on the mechanism details. We think that the definition of when, where, and how to use Transfer Learning should be called into question when it comes to Human Activity Data. Therefore this paper concentrates more on understanding the source and target datasets, as well as understanding the process of weight transfer between models. We want to encourage researchers to take a closer look at these aspects and dig deeper into the mechanism of Transfer Learning.

5.1.2. Methodology

Two publications that influenced our choice of datasets are [76] and [146]. The results presented here show, that Skoda Mini Checkpoint is basically suitable as a source dataset and PAMAP2 as target data. PAMAP2, on the other hand, has already proved in previous publications, for example, [159] or [82], to be suitable for use with neural networks.

Datasets. We have chosen to evaluate these two publicly available activity recognition datasets as the type of sensors, the sampling rate, and the location at which the sensor was worn match particularly well:

PAMAP2: The PAMAP2 dataset consists of 19 different classes of activities of daily living and is recorded with a sampling rate of 100Hz and a sensitivity of ±16g. 9 subjects participated in the experiment. To train the PAMAP2 models we used data that has been recorded following the experiment protocol. We also concentrate on activities that are performed by every subject. With these conditions, the used data is reduced to 7 subjects performing 8+1 (null class) different activities of daily living. Activities that are taken into account are: null (0), lying (1), sitting (2), standing (3), walking (4), ascending stairs (12), descending stairs (13), vacuum cleaning (16) and ironing (17).

Skoda Mini Checkpoint: The Skoda Mini Checkpoint dataset is recorded by 1 subject, performing 10 different activities, with a sampling rate of approx. 98Hz, and a sensitivity of ±3g. Classes used from this dataset are restricted to the ones, where the activity is performed equally by both hands. Hence classes that were taken into account are null (32), open hood (49), close hood (50), check gaps on the front door (51), close both left doors (54), check trunk gaps (55), open and close trunk (56) and check steering wheel (57).

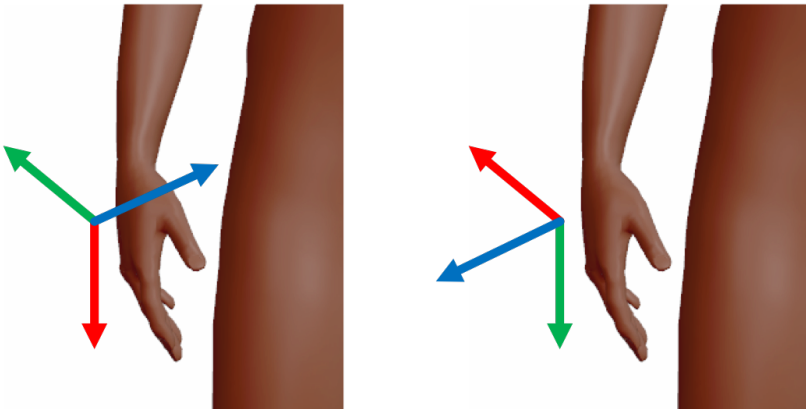


Figure 5.1 Default Sensor Orientation of PAMAP2 (left) and Skoda Mini Checkpoint (right). The X-axis (red) and Y-axis (green) are switched and the Z-axis (blue) is inverted.

While corresponding with the authors of PAMAP2 and Skoda Mini Checkpoint we realized that these two datasets were recorded with different sensor orientations. Figure 5.1 illustrates this problem.

Preprocessing. We used the same preprocessing steps for the baseline model and the transferred model. **(1)** concatenate the data into an array with one channel per sensor-axis, **(2)** delete all synchronization gestures from the dataset, **(3)** scale all axes of the data at ones between -1 and 1, **(4)** apply a jumping window with a length of 50 samples and an overlap-ratio of 50%, **(5)** shuffle the windows with a fixed random seed. For defining the label of the current window we followed the approach used in [159], where the label of the last sample defines the label of the window. Early tests showed, that the classification results between the default 98Hz and resampling to 100Hz for the Skoda Mini Checkpoint dataset are marginal and therefore negligible.

Baseline Model. In order to investigate the effects of transfer learning between different types of sensors, sensors mounted on different body parts, as well as misaligned axes, we had to train two different baseline models. One trained on the wrist-worn PAMAP2 accelerometer, and one on the Skoda Mini Checkpoint, using only the data from the accelerometer of the right wrist. Instead of using RMSProp as the

Table 5.1 Parameters used for the baseline model as well as the different transfer methods. Fixed parameters for all models are Batch-Size (64), Conv. Kernel-Size (5x3), LSTM-Cells per layer (128), learning-rate (0.001). After the transfer, the trainable parameters were either: frozen (f), trainable (t), or reinitialized and trainable (lecun_uniform)

Parameter	**Baseline Model**	**Transfer Method 1**	**Transfer Method 2**	**Transfer Method 3**	**Transfer Method 4**
Training Epochs	1000	30	30	30	30
Weight Init. Conv.-Layers	lecun uniform	pretrained (f)	pretrained (f)	pretrained (f)	pretrained (t)
Weight Init. LSTM-Layers	lecun uniform	pretrained (f)	pretrained (t)	lecun uniform	lecun uniform
Optimizer	Adadelta	RMSProp	RMSProp	RMSProp	RMSProp

optimizer, as proposed by [159], we switched to Adadelta, which performs slower but is more stable. RMSProp showed an unstable behavior regarding the classification performance with massive negative peaks in longer training periods but seems to be a good choice for fine-tuning operations. All training parameters are listed in Table 5.1.

Transfer Learning. We have applied four different methods of transferring the model. All methods follow the weight transfer method, e.g. used in [125] and [146], to transfer the pre-trained model. We transferred the pre-trained weights from the baseline model and replaced the classification layer with an untrained one, which fits the number of classes of the target dataset. We also switched the optimizer of the transferred model to RMSProp, since we only fine-tuned our model for 30 epochs. Following we distinguish between different levels of post-transfer trainable layers: **(1)** All layers are frozen after transfer, except the classification-layer, **(2)** Only the ConvBlocks are frozen after transfer, LSTM-Layers stay trainable, **(3)** the ConvBlocks are frozen after transfer, LSTM-Layers stay trainable, but are reinitialized with lecun-uniform initialization and **(4)** the Conv.- and LSTM-Layers are trainable, but LSTM-Layers are reinitialized with lecun-uniform initialization. Figure 5.2 depicts the used architecture and transfer method. To evaluate the results, we determined the respective Training F1-Score. In order to simulate all possible orientations of a sensor relative to the baseline model, we decided to permute and invert the position of the sensor axes. This results in 48 possible combinations. Thus, all models were transferred 48 times in each test that was not transferred back to the source dataset. A transfer back to the source data was done as a sanity check. These sanity checks, as well as transfer within the dataset, but to another sensor worn at the same position, are done with a leave-one-fold-out cross-validation with 4 folds.

5.1.3. Results and Evaluation

We tested Transfer Learning between different sensor locations, different sensor types, and different sensor orientations for intra-, as well as inter-dataset transfer. Transfer back to the source dataset was performed as a sanity check, see experiment **(1)**, and **(5)**. Similar results after transfer with method 1 ensure that no errors occurred during transfer or data preprocessing. If irregularities occur in the preprocessing or transfer process, the after-transfer performance of method 1 would decrease significantly. The result of **(1)** with methods 2, 3, and 4 shows that transfer learning harms the classifier in general and needs to be fine-tuned to perform reliably. The result of experiment **(2)** with method 1 must be subjected to closer examination. The best result is achieved after the X-axis is first inverted and then swapped with the Z-axis, which results in a Z, Y, -X orientation. Whether this orientation corresponds to the actual position of the axes relative to the sensor worn on the wrist cannot be said with certainty at this point, but it is evident, that this result deviates from the average by about 10%. Remarkable is experiment **(3)** with method 1, in which the model trained on the data of the right wrist of Skoda Mini Checkpoint, is applied to the data of the left wrist. The best result was obtained after leaving the axes in the default position but inverting the X-axis. Thus the left-hand data was artificially mapped to the orientation of the

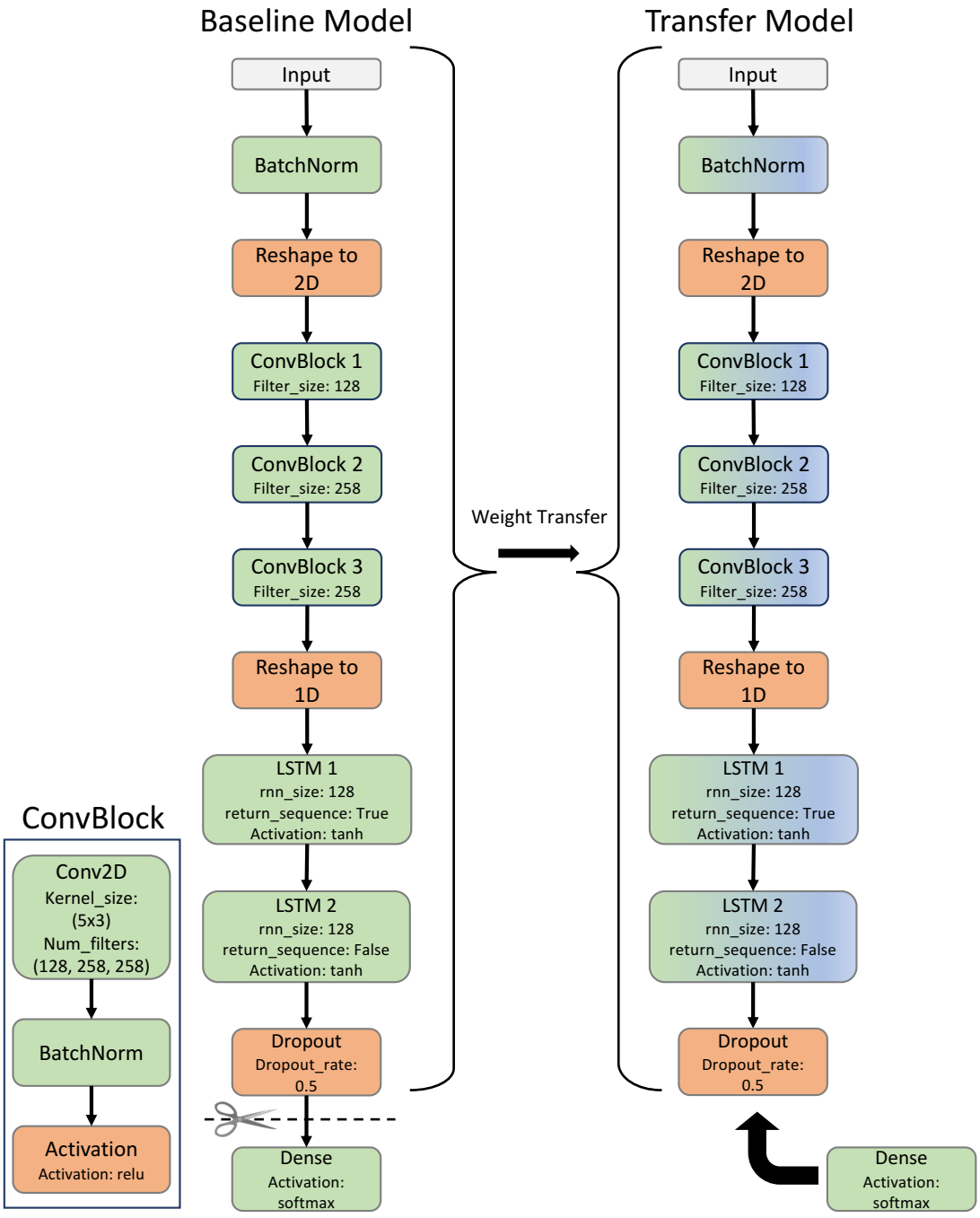


Figure 5.2 DeepConvLSTM [159] architecture. Red blocks do not have trainable parameters, whereas green blocks are trained, transferred, and frozen (represented as blue blocks) in the target model, depending on the used transfer method. The original last dense layer is replaced with a new output layer during transfer, with a size according to the number of classes of the target dataset. One ConvBlock consists of three layers, a convolutional layer, a batch normalization layer, and an activation layer with a ReLu activation function.

Table 5.2 Deep Learning train-F1-Score in %, given as minimum, maximum, and mean. PAMAP2 (P), Skoda (S), Accelerometer (A), Gyroscope (G), Magnetometer (M), Wrist (W), Chest (C).

	Source → Target	Method 1			Method 2		
		Min	Max	Mean	Min	Max	Mean
1	**S (W, Right) → S (W, Right)**	97.5	98.5	97.8	93.1	93.5	93.2
2	**S (W, Right) → S (C, Right)**	43.3	64.4	54.8	84.1	93.6	93.0
3	**S (W, Right) → S (W, Left)**	50.8	61.0	54.6	89.1	93.4	91.1
4	**S (W, Right) → P (A, W)**	12.1	26.6	20.7	54.9	69.4	63.6
5	**P (W, A) → P (W, A)**	82.3	82.5	82.4	92.8	93.0	93.0
6	**P (W, A) → P (C, A)**	23.3	42.6	35.2	59.1	73.4	65.8
7	**P (W, A) → S (W, Right)**	38.1	46.3	42.8	73.0	86.4	79.4
8	**P (W, A) → P (W, G)**	03.2	03.4	03.3	21.6	23.6	22.4
9	**P (W, A) → P (W, M)**	36.6	37.4	37.1	66.6	68.1	67.5
	Source → Target	**Method 3**			**Method 4**		
1	**S (W, Right) → S (W, Right)**	96.0	96.6	96.2	60.9	64.2	63.0
2	**S (W, Right) → S (C, Right)**	85.5	94.4	89.0	42.0	69.8	63.4
3	**S (W, Right) → S (W, Left)**	90.1	94.4	92.0	63.4	77.9	72.6
4	**S (W, Right) → P (W, A)**	54.5	70.7	64.6	11.0	63.7	52.1
5	**P (W, A) → P (W, A)**	94.7	95.4	94.9	54.9	55.9	55.1
6	**P (W, A) → P (C, A)**	59.3	73.5	66.7	57.1	64.4	60.7
7	**P (W, A) → S (W, Right)**	77.3	87.7	82.3	34.2	72.8	61.5
8	**P (W, A) → P (W, G)**	21.5	23.2	22.5	05.4	07.1	06.0
9	**P (W, A) → P (W, M)**	67.6	68.0	67.5	09.8	19.8	15.5

trained model. Experiment **(6)** resulted in the highest F1-Score when the axes were left in the original position but differed by up to 19.3% during the permutation test. This means that the alignment of the chest sensors matches the alignment of the one worn at the wrist. Inter-Dataset Transfer Learning, experiment **(4)** and **7** with method 1 always resulted in very low F1-Scores. However, the transfer from PAMAP2 to Skoda **(7)** had better results than from Skoda to PAMAP2 **(4)**, which might be a direct result of the bigger size and variability of the PAMAP2 dataset. A transfer between types of sensors is in general not recommendable. **(8)** and **(9)** show that a model trained on accelerometer data is not capable of classifying the same activities recorded by a gyroscope or magnetometer.

5.1.4. Discussion and Conclusion

We started this experiment by assuming that by properly adjusting the position and orientation axes of the inertial data along the sensor axes we could significantly increase the classification results. We could demonstrate this with the results of method 1, but these results did not reach the significance as initially expected and are therefore not an acceptable final state for a classifier. Matching the alignment of the sensor axes results in a more adapted classifier, but it is not possible to achieve the

classification properties of the baseline model. Due to the mostly frozen architecture, the adaptation process of fine-tuning the classification layer reaches its limits very quickly.
The results of methods 2 and 3 are very similar. However, these experiments show that it is basically advisable to reinitialize the LSTM-Layers to default since the F1-Score is on average 3.4% higher with method 3 than with method 2. The experimental results of method 4 demonstrate that convolutional layers should not be fine-tuned after model transfer. The comparatively worse results of this method are caused by the outputs of the convolutional layer being fed as input to the LSTM-layers due to their position in the architecture. By re-initializing these layers, but also keeping the convolutional layers trainable, the pre-trained data-dependent link between these layers is lost.
The datasets used in this paper share many modalities, such as the position of the sensors on the body, the sampling rate, and the sensor technology used, but differ fundamentally in the underlying classes. Thus, we assume that the features of the filters trained in the convolutional layers are very dataset-dependent and thus class-specific. Using pre-trained weights can provide a speed advantage and thus lead to faster network convergence, due to less trainable parameters, but we consider the impact on the final classification performance, even with artificial adjustments of the orientation and position of the sensors, to be marginal. These results largely correspond to those of [146].

It is surprising that although the modalities of both datasets are largely identical, a transfer between them is always accompanied by strong performance losses. This observation leads us to the following research challenges, which we leave open at this point for the research community to address:

(a) Under which exact conditions is Transfer Learning recommendable for wearable-based activity data?
(b) How transferable are the pre-trained convolution filters between inertial activity datasets?
(c) Which preprocessing steps are suitable to make models transferable, regardless of their architecture?

We argue that the three questions for designing transfer learning – What? When? How? – are hard to adapt from other disciplines and should be reconsidered for Transfer Learning with inertial sensors-based signals.

Section 5.2

Data Augmentation Strategies for HAR

[94] Hoelzemann, Alexander, Nimish Sorathiya, and Van Laerhoven, Kristof
Data augmentation strategies for human activity data using generative adversarial neural networks
2021 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops)
https://doi.org/10.1109/PerComWorkshops51409.2021.9431046

Portions of the original publication have been removed or edited for inclusion in this thesis. However, no changes were made that altered the results or conclusions presented in the original work.

Contributions:

- All authors designed the study.
- Nimish Sorathiya and I implemented the study and analyzed the results.
- Kristof Van Laerhoven guided this work and assisted in the methodologies.

Previous studies have shown that available benchmark datasets from the field of Human Activity Recognition are of limited use for Deep Learning applications. This can be traced back to issues in the quality, the scope, as well as in the variability of the datasets. These limitations often lead to the overfitting of networks and thus to results that are only conditionally generalizable. One way to counteract this problem is to extend the data by using data augmentation techniques. This paper presents an algorithm and compares two augmentation strategies: (1) user-wise augmentation and (2) fold-wise augmentation, to expand the size of a dataset with any number of synthetic samples. This is demonstrated using the PAMAP2 dataset. These synthesized data resemble the user- and activity-specific characteristics and fit seamlessly into the dataset. They are created by a recurrent Generative Adversarial Network, with both the generator and discriminator modeled by a set of LSTM cells to produce the synthetic time-series data. In our evaluation, we trained four DeepConvLSTM models with supervised learning, three times with a LOSO cross-validation: one baseline model and two times with additional data but different augmentation strategies, as well as one model without cross-validation that monitors the synthesized data quality. The compared augmentation strategies demonstrate the impact as well as the generalized nature of the augmented data. By increasing the size of the dataset by factor 5, we improved the F1-Score by 11.0% with strategy (1) and 5.1% with strategy (2).

5.2.1. Introduction

In the context of deep learning and neural network development, having sufficient data to train and test algorithms is crucial for obtaining a high-performance classification model. However, this essential step is often hindered by the fact that not enough data is freely available for many types of applications. For this reason, methods for data synthesis have been developed in the past years. These algorithms are already widely used in computer vision and natural language processing, however, they are still in their early stages of development with regard to activity recognition from wearable inertial sensors. In order to accelerate research in the field of Deep Learning for Human Activity Recognition, it is essential to increase the scope of its public datasets in the future. We already know from other disciplines that more data can lead to more precise results and is an effective tool against overfitting. The further development of augmentation and synthesis algorithms can serve as a catalyst that will enable us to close the current gap in the available data. In this paper, we propose a neural network architecture based on [67] and further developed to generate data for an arbitrary number of samples of sensor-based activity data. The network can be trained to synthesize both subject- and activity-specific characteristics. The quantitative enlargement of the dataset improves the classification potential of the neural network on the one hand and protects it efficiently against overfitting on the other hand. Our tests show that these artificial data can be used to increase the classification capabilities of a neural network model, due to an increase in variability and scope of the dataset, but the impact varies depending on how the data was synthesized and merged back into the initial dataset.

5.2.2. Experiment

The PAMAP2 dataset consists of 19 activities of daily living and was recorded by 9 subjects. The sampling rate of the dataset is 100Hz and the sensitivity $\pm$16g, the sensors are placed on the chest, right ankle, and right wrist [173]. For our experiment, we decided to use the protocol subset, since these data were recorded according to a fixed protocol sequence and therefore can be interpreted more uniformly. Furthermore, we limited the subset to the wrist sensor and to activities that are recorded by each subject equally. We decided to not take the null class into account, following the author's recommendation. Under these conditions, the data is reduced to a subset that contains 8 subjects performing 6 different activities of daily living. Activities that are taken into account are: lying, sitting, standing, walking, vacuum cleaning, and ironing.

Our developed approach is depicted as a process cycle and is easy to follow up, see Fig. 5.5. This figure a total of 3 different variations of the dataset that are used or created during the augmentation process. One is the initial dataset, as earlier described. The same data, but organized as Leave-One-Subject-Out (LOSO)-folds is further referred to as α-dataset, which consists of all selected subjects and activities. The α-subset is used to obtain the ground truth, also called baseline, and serves as

the input for synthesizing new data. Since the used protocol subset of PAMAP2 contains data from 8 different subjects, our α-subset contains 8 folds, wherein each of the subsets one subject is excluded and used as the test data. After the augmented data is merged into the α-dataset, the set is referred to as β-dataset.

Network Architectures: DeepConv-LSTM. A DeepConv-LSTM architecture is used, see Ordoñez et al. [159], to train four different models. The first model is trained with all subjects and all activities used in this experiment. This network monitors the quality of the generated samples by predicting the sample classes. Another model is trained by using the LOSO cross-validation with the α-subset to obtain the baseline for the final evaluation. A third and fourth model is trained with the β-subset, which contains the LOSO-folds (α-subset) as well as the synthesized data. However, the β-subset differs depending on the chosen augmentation strategy.

Network Architectures: Generative Adversarial Network (GAN). The architecture of this work is based on the network introduced as Recurrent GAN [67] and follows an architecture in which both, the generator and discriminator, are LSTM-Networks instead of multi-layer perceptrons. The generator network takes random noise at the start of the training. The length of the noise-vector corresponds to the number of timesteps of the LSTM-cell. The discriminator network is used as a binary classifier, which takes the output from the generator as synthetic time series and real data at each LSTM timestep.

Training of the discriminator as a binary classifier minimizes the average negative cross-entropy between the prediction and real labels for both synthetic and real examples. Considering CE as the average cross-entropy between sequences X_n and y_n, where X_n ($X_n \in \mathbb{R}^{T*d}$) is the matrix that comprises the output sequence T from the LSTM cells in the discriminator and y_n, where y_n can be a vector, 1 or 0; the discriminator loss is given by,

$$D_{loss}\left(X_n, y_n\right) \;=\; -CrossEntropy(\mathrm{LSTM}_D(X_n), y_n) \tag{5.1}$$

This loss is used by the generator to mislead the discriminator network by producing real-like data that minimize the average negative cross-entropy between the discriminator output on synthetic data and the actual label, considering Z_n as a sequence of samples from noise space z;

$$\begin{aligned} G_{loss}\left(Z_n\right) \;&=\; D_{loss}(\mathrm{LSTM}_G(Z_n), 1) \\ =\; -CrossEntropy(\mathrm{LSTM}_D(\mathrm{LSTM}_G(Z_n)), 1) \end{aligned} \tag{5.2}$$

The generator consists of LSTM cells with 100 units as hidden layers and a linear activation function, instead of the tanh-function used by [67]. The discriminator uses the sigmoid-function as the output activation function. Both, the generator and discriminator are simultaneously trained at each epoch. Considering the data generating distribution as p_x from the generative distribution as p_g; After an arbitrary

number of timesteps, both of the networks will hold an equilibrium condition and cannot be further improved than p_g=p_{data}. The architecture is depicted in Figure 5.3.

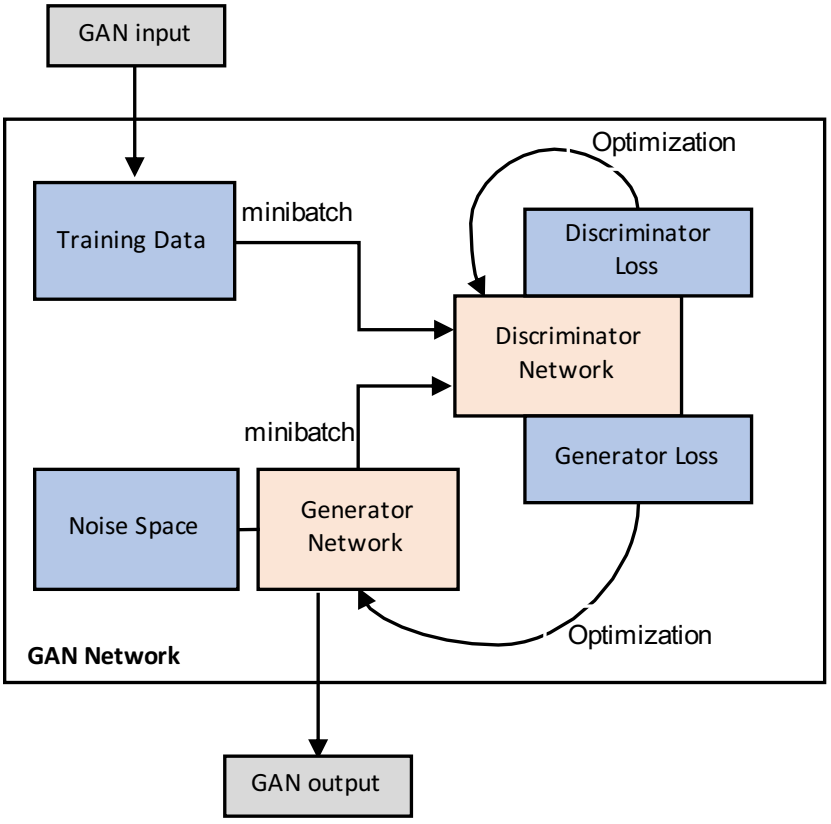


Figure 5.3 GAN architecture. Both networks, Generator and Discriminator (orange boxes), consist of 100 hidden LSTM cells. Random noise from noise space is fed as input to the Generator. The trained Generator synthesizes samples as an output. Using cross-entropy loss, both networks are optimized at each time-step. The network will take the original sample windows from α-subset as an input and generate samples (β-subset) as the output of the network.

Methodology. In our experiment, we first trained a model of the DeepConv-LSTM with the complete protocol-subset of PAMAP2. This network is trained for 200 epochs and achieved a validation F1-Score of 96%. This means that this network knows the characteristics of all subjects and all used activities, therefore it is able to distinguish between real and fake data and can be used as a model to select just appropriate data generated from the GAN. Important training parameters for the GAN network are: learning-rate = 0.10, batch-size = 20, latent-dimension (or noise space for generator input) = 10, and the number of times the generator and discriminator network are optimized at each epoch = 5. Due to the unknown number of exactly needed training epochs of the generator and discriminator, in which the GAN starts to produce real-looking synthetic data, the network needs to be trained for many epochs. Training the GAN for approximately 1000 epochs is an appropriate estimate to start the process. As soon as fitting hyperparameters and epochs are found, we are able to synthesize an arbitrary number of samples.

Since we are tuning between two networks (generator and discriminator), the discriminator often shows lower loss values than the generator. Although the generator misled the discriminator, the produced data does not look realistic. Therefore the

synthesized data from the generator is fed for a quality-check to the DeepConv-LSTM. If the F1-Score achieves $\geq 95\%$, the data is considered to have reached the supposed quality and the data will be saved or otherwise discarded. If the required amount of data is reached, the process will be terminated.

Two different strategies, (1) Subject- & Activity-Wise Augmentation and (2) Folder- & Activity-Wise Augmentation, are developed, see Fig. 5.4. Both strategies follow the process cycle as shown in Fig. 5.5, but differ in the input data of the GAN. Therefore

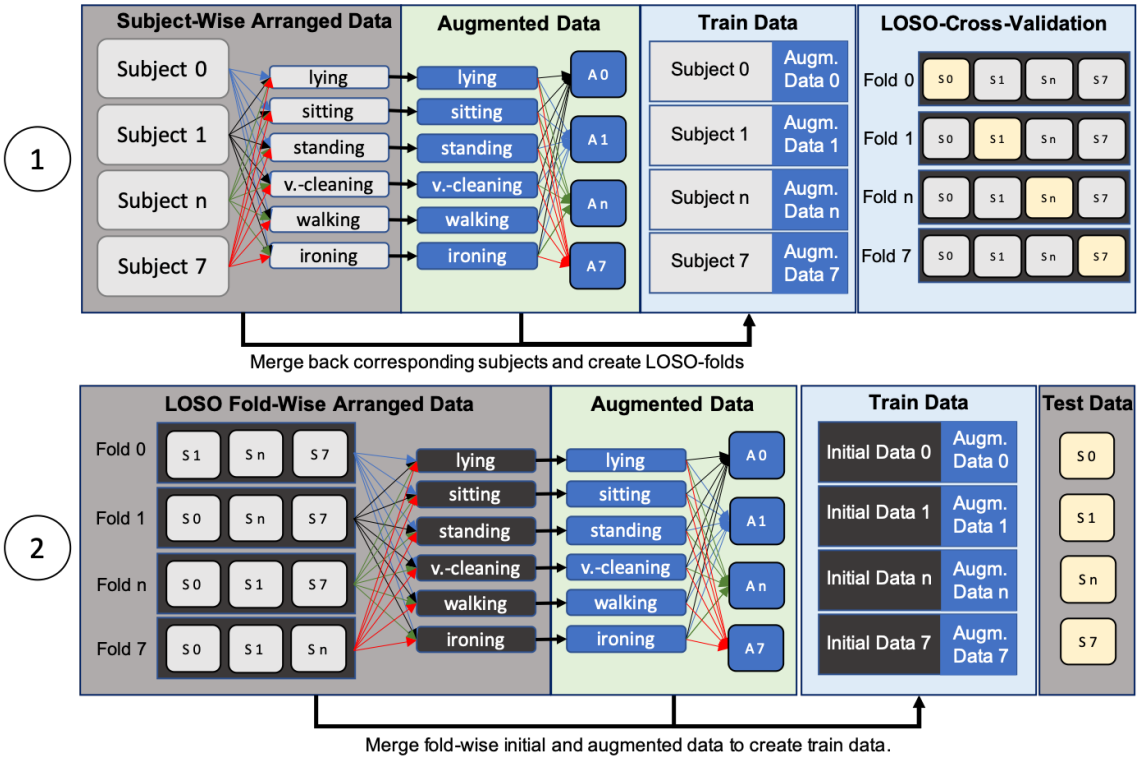


Figure 5.4 Data Augmentation Strategies: Grey background represents the initial dataset, green the augmented data, and blue the data after merging both (β-dataset). Yellow squares represent the test-subject for each fold. (1) Subject- & Activity-Wise augmentation and merging strategy. After augmenting the data, the augmented data is merged back into the subject's data, afterwards the LOSO-Folds will be created; (2) Fold-Wise augmentation and merging strategy. LOSO-Folds are created before the augmentation process starts. The fold-wise arranged data is then used as input for the augmentation. The synthesized data results in non-subject-specific data, since it contains characteristics from all subjects of the fold.

the augmented data show deviant characteristics.

(1) Subject- & Activity-Wise Augmentation

This strategy uses only the personal activity of a subject as input. The data generated in this way is thus subject-specific. The synthesized data is then added to the subject in the original dataset. Afterward, the different folds for the cross-validation are generated.

(2) Fold- & Activity-Wise Augmentation

The data augmentation process itself is divided into 2 phases: (1) Generator phase and (2) Discriminator phase. Figure 5.5 illustrates the complete augmentation process. A separate GAN must be trained for each activity. If the generated samples cannot be distinguished from real samples anymore, they will be saved, otherwise, they will be discarded. Generator and Discriminator networks train parallel on the real data. This process is repeated till the discriminator unit decides that the data looks real. Therefore we talk about this as a cycle with n iteration steps. However, due to improper parameter tuning of the generator and discriminator, it can happen that the generator misleads the discriminator, which results in unreal-looking samples.

The fold-wise activity selection uses activity data of all subjects from a fold as input

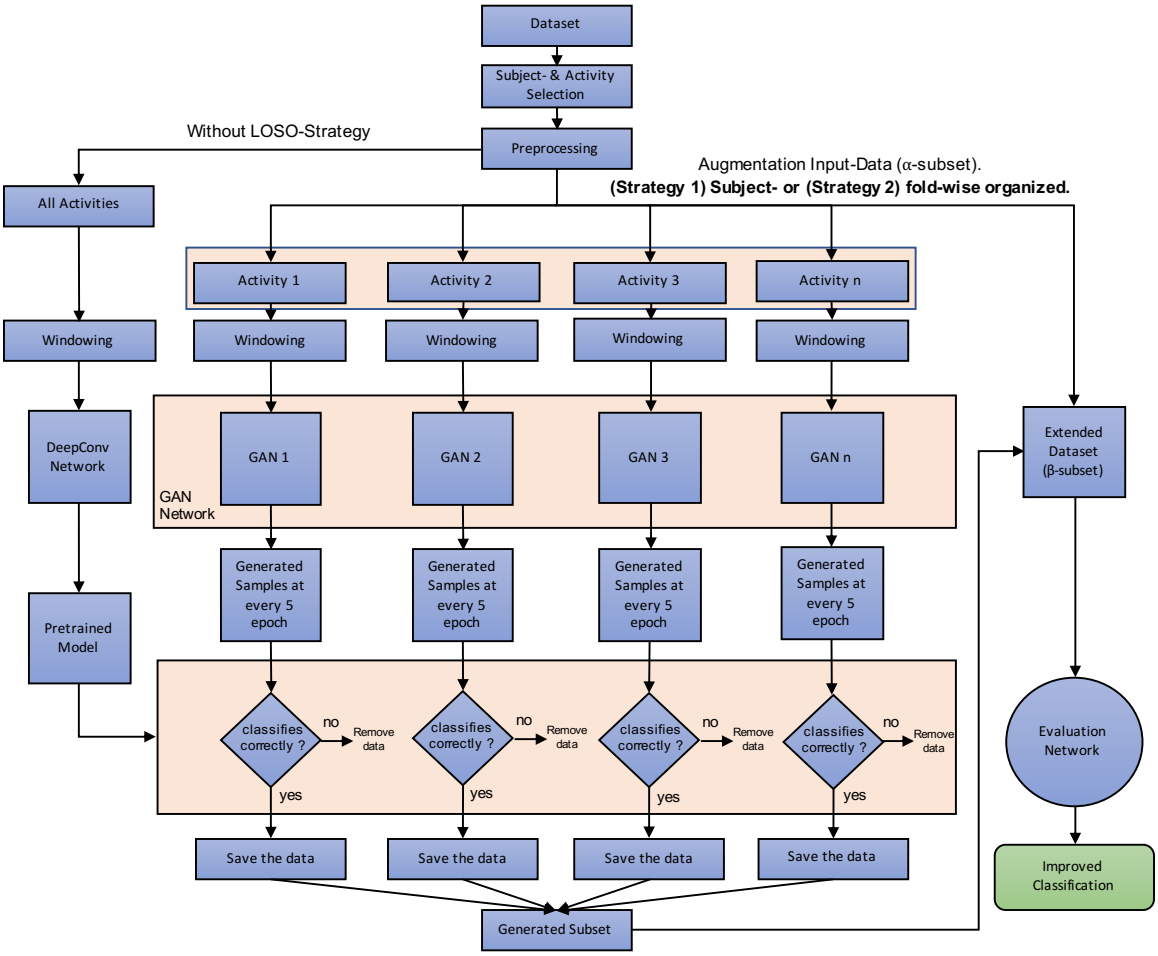


Figure 5.5 Data Augmentation Process Cycle: The complete dataset is needed to train the model that monitors the quality of the augmented data. The α-subset represents the input data, the β-subset the dataset after merging the augmented data with the α-subset. For every activity, subject- or fold-wise organized data, a new GAN needs to be trained. After 5 epochs the generated data is tested, if it reaches the predefined F1-Score of 95% it will be kept. If not, the data will be discarded.

for the augmentation. The resulting data can no longer be assigned to a specific subject. Rather, it contains characteristics of each subject in the fold. Thus, the data is not assigned to the subjects but is merged into the folds directly. In contrast, the test dataset is not enlarged as in strategy (1), instead, the test subjects of the α-dataset are used. Once the subjects and activities to be augmented have been selected, the preprocessing is applied. It is important to note that the data generated by the GAN will be of the same nature as the input data. This means that if raw data has to be generated, the preprocessing must not include operations that alter the raw data itself. Our goal is to produce raw data that appears genuine. Therefore, during preprocessing, we only remove missing values from the dataset and do not apply normalization, even though normalization leads to better classification results. Afterwards, a jumping window algorithm is applied on both subsets with a window length of 100 samples (1 second in time-domain) and without overlapping samples. The windows are labeled according to the method proposed by [159], where the assigned label of the window is identical to the last sample of the window. These labels are one-hot encoded with 0.0 or 1.0. After merging synthesized and original data. Our final subset is called β-subset.

We synthesized 80000 samples per class using this method, which is approx. 5 times more than the original dataset. Table 5.3 sums up the process depicted in Figure 5.5 in a compact format and can be used as a progress guide to implementing a data augmentation algorithm for sensor-based human activity data.

5.2.3. Results

We trained our classification network with both augmentation strategies and compared the results to the baseline, the results are summed up in Table 5.4 and visualized as confusion matrices in Figure 5.6. As shown in the table, the strategies can increase

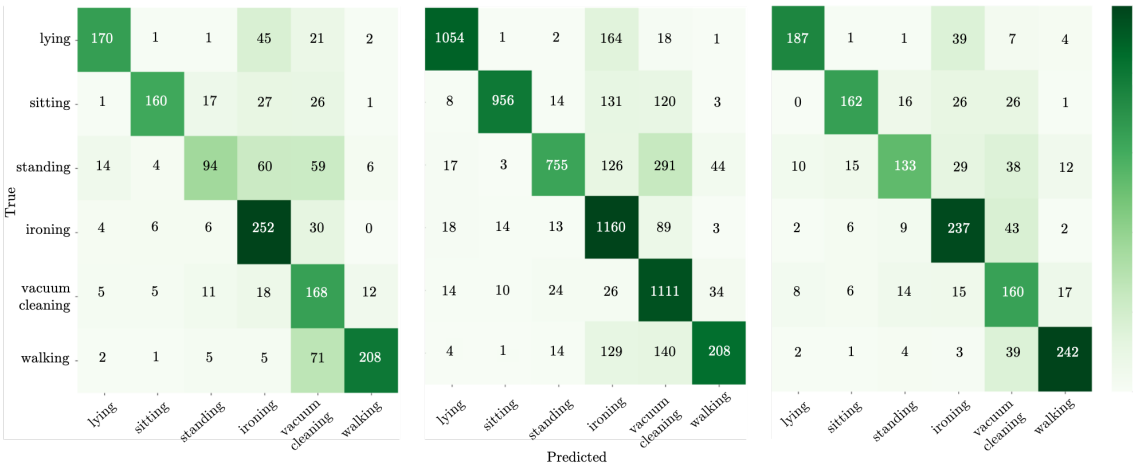


Figure 5.6 Confusion Matrices of the average classification results from the loso-cross-validation. From left to right: without augmented data, with Activity- & Subject-Wise augmented data, with Fold- &Activity-Wise augmented data.

Table 5.3 Process guide to augment data exemplary on PAMAP2.

Step	Action	Result	Pitfalls
(1) Subject and Activity Selection	(1) Select Subjects (2) Select Activities	protocol subset	Select Activities or Subjects with insufficiant number of samples.
(2) Preprocessing	(1) Delete missing values (Optional) Normalization (2) Create Windows (3) One-Hot-Encoding of labels (4) Create LOSO-Subsets	α–subset (Preprocessed)	If the data is normalized, be sure that the same normalization method is applied on all subsets. We recommend to skip normalization and work with raw data.
(3a) Train Monitoring-Network	Use protocol subset to train the test-network	trained model to test the quality of the augmented data	Over- or underfitting of the Network. If an over- or underfitting of your network already happens with the complete dataset, it will also happen with the reduced β-subsets. **Hint:** Quality-check with cross-validation on the baseline model, to see if the model is over- or underfitted.
(3b) Train baseline	Calculate the baseline by training a DeepConv-LSTM with the α-subset	LOSO-Baseline	
(3c) Train GANs	(1) Select subjects and activities, for which data should be generated. (2) Decide for an augmentation strategy (3) Train the GAN-Networks and generate augmented data using α-subset	Augmented windoows of samples.	The Generator does not produce realistic samples at initial steps although discriminator loss is quite low. Introducing a new activity, results in fine-tuning the parameters at first
(4) Merge Data	(1) Merge α-subset with augmented data	β-subset	
(5) LOSO cross-validation	(1) Train/Test Model with β-subset and LOSO-Cross-Validation	Final classification results	Not choosing the correct metrics with respect to the dataset attributes.

Table 5.4 Precision (P), Recall (R), and F1-Score (F1) as resulted from the different augmentation strategies for every activity class, as well as the calculated weighted average. The baseline without data augmentation reached an F1-Score of 67.5%. Both augmentation methods improved the classification results. Subject-specific augmentation method improves the total F1-Score to 78.5%. The Fold-Wise augmentation pushes the F1-Score to 72.6%.

Fold	0			1			2			3			4			5			6			7		
Metrics	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Total avg F1-Score: 67.5	Baseline without Data Augmentation																							
lying	76	36	49	100	92	96	100	91	95	94	72	81	83	93	88	78	94	86	100	95	97	02	00	01
sitting	00	00	00	99	59	74	90	98	94	95	95	95	91	76	83	99	83	90	99	91	95	68	52	59
standing	00	00	00	20	11	14	97	54	70	79	80	80	50	38	43	84	36	50	88	92	90	19	02	03
ironing	31	100	48	54	95	69	84	71	77	94	93	94	78	89	83	80	89	84	89	97	93	41	49	45
vacuum cleaning	60	79	68	48	83	61	53	89	67	51	64	56	47	60	53	64	89	74	79	73	75	21	80	33
walking	82	84	83	99	50	67	95	93	94	86	88	87	83	72	77	93	95	94	96	97	96	44	01	03
weighted avg	42	50	41	70	64	63	87	83	84	84	83	83	73	73	73	83	81	80	92	92	92	33	30	24
Total avg F1-Score: 78.6	Subject- & Activity-Wise Augmentation																							
lying	96	95	95	100	98	99	99	98	99	99	92	95	96	97	97	95	99	97	99	100	100	01	00	01
sitting	00	00	00	99	96	97	95	100	97	98	97	98	98	93	96	100	97	98	99	99	99	85	40	54
standing	69	06	11	94	91	93	95	90	92	93	98	95	81	52	63	94	52	67	94	99	96	08	00	00
ironing	29	100	45	98	98	98	96	85	90	98	97	98	88	97	92	98	96	97	97	98	98	33	46	38
vacuum cleaning	78	89	83	84	95	89	86	96	91	89	92	90	53	89	67	67	91	77	95	90	93	28	88	42
walking	55	16	25	99	94	96	95	97	96	97	96	96	91	60	72	88	98	93	98	97	97	86	61	71
weighted avg	55	51	44	96	95	95	94	94	94	96	95	95	85	81	81	90	89	88	97	97	97	40	39	35
Total avg F1-Score: 72,6	Fold- & Activity-Wise Augmentation																							
lying	81	70	75	100	92	96	98	92	95	98	86	92	98	92	95	83	94	89	95	98	97	02	00	01
sitting	00	00	00	90	59	72	87	100	93	94	95	95	93	72	81	80	94	86	97	92	95	59	51	54
standing	08	00	01	74	95	83	92	67	77	85	91	88	48	42	45	83	46	59	80	97	88	43	02	05
ironing	33	100	49	93	91	92	83	57	68	97	92	94	81	87	84	94	82	88	89	97	93	35	38	37
vacuum cleaning	76	73	74	59	79	67	56	86	68	65	75	70	67	60	63	63	86	73	92	48	63	23	79	35
walking	89	86	88	97	83	90	92	93	93	91	89	90	68	91	78	92	94	93	90	97	93	84	37	52
weighted avg	49	56	49	87	84	84	85	83	83	89	88	89	76	76	75	84	83	82	90	89	88	42	35	31

the F1-Score by about 5.1% using strategy 2. The subject-specific augmented data, strategy (1), increases the F1-Score by 11.0%. Furthermore, the cross-validation shows that the characteristics of the data of subjects 0 and 7 do not seem to match those of the other study participants, resulting in lower classification results. They were only conditionally increased for the classes lying, standing, vacuum cleaning, and walking (only strategy 2). The confusion, visible in Figure 5.6, belongs mostly to these subjects and shows that subject-specific confusion is not solvable by just increasing the number of samples, since even though the number of samples was increased by factor 5, and the confusion remained. The baseline results of our experiment do not reach results presented in other papers, for instance, [238], [82] that worked with similar architectures and datasets. This is due to the fact that we have limited ourselves to the wrist sensor, as well as the smaller protocol subset, and have refrained from preprocessing.

5.2.4. Conclusions and Discussion

This section introduces a new approach to augment sensor-based human activity data. The generative part of our architecture works with a Generative Adversarial Network (GAN), which builds on the work of Esteban et al. [67] and is further developed for using inertial data. Our method synthesizes data that mimics the input data characteristics. By following two strategies, we are able either to augment subject- or fold-specific activity data. The GAN is able to produce raw data, as well as preprocessed appearing signals. We argue that with the generated data we are able to increase the scope and variability of a dataset, which helps to increase the classification performance of a neural network and to prevent negative effects, such as over- or underfitting. This work has also shown that adding augmented data could have negative effects on certain classes and subjects. This approach is applicable to an arbitrary number of activities and subjects and can be transferred to other sensor-based human activity datasets. Through the presented process cycle, we offer an easy-to-follow method that helps other scientists adopt, reproduce, and integrate such a method into their own experiments. The architecture of the test network can be exchanged at will and thus be adapted to individual needs. However, further development of GAN architectures is necessary to be able to overcome the time-consuming disadvantage of choosing the correct hyperparameters. To avoid this factor in the future, we plan to extend the architecture with an independently acting search algorithm to find satisfying hyperparameters for the GAN. In further experiments, it is important to consider how much the size of the augmented dataset influences the classification capabilities of a model. Due to space constraints, those effects were not explored in detail, but they are an important factor for the real-world application of such strategies and algorithms.

Section 5.3

Summary

Section 5.1 is dedicated to investigating the applicability of transfer learning techniques in the context of sensor-based human activity data analysis. To explore the potential and limitations of this approach, a series of experiments were conducted. These experiments encompassed endeavors such as sensor alignment optimization as part of preprocessing, model transfer between different sensor locations, and model transfer between distinct datasets. We distinguished between different levels of post-transfer trainable layers: (1) All layers are frozen after the transfer, except the classification-layer, (2) Only the ConvBlocks are frozen after the transfer, LSTM-Layers stay trainable, (3) the ConvBlocks are frozen after the transfer, LSTM-Layers stay trainable, but are reinitialized with lecun-uniform initialization and (4) the Conv.- and LSTM-Layers are trainable, but LSTM-Layers are reinitialized with lecun-uniform initialization. The findings, summarized in Table 5.1.3, reveal that the effectiveness of transfer learning is contingent upon specific model configurations and dataset characteristics. Transfer learning exhibits advantages primarily in terms of reduced training durations and a diminished requirement for extensive data to adapt a model. Nevertheless, the overall improvement in a model's capacity to accurately classify data was not found to be statistically significant. Specifically, the attempt to transfer a model between datasets originating from distinctive domains yielded unremarkable results, as the classifier's performance showed no noticeable enhancement post-model transfer.

Section 5.2 or [94] introduces a novel approach to augment sensor-based human activity data using a custom GAN architecture designed for inertial data. The approach offers two strategies for augmenting subject-specific or fold-specific activity data, with the aim of expanding dataset variability and scope. The generated data is intended to improve neural network classification performance while addressing overfitting and underfitting issues. I highlight, that the introduction of augmented data may have negative impacts on specific classes and subjects. The approach is versatile and can be applied to various activities and subjects, making it accessible to other researchers. I further acknowledge the time-consuming process of hyperparameter selection for the GAN and suggest future work to automate this task. Additionally, I emphasize on the need for further investigation into how the size of the augmented dataset affects classification performance, especially in real-world applications. Overall, this paper contributes to the exploration of data augmentation strategies with potential implications for sensor-based human activity data analysis.

Chapter 6

Conclusion and Future Work

Accurate and robust human activity recognition relies heavily on high-quality datasets. However, several factors can introduce noise, biases, and errors into activity recognition datasets, which then propagate through the modeling pipeline. One critical issue is the quality of manual annotations for sensor data, which are required to train and evaluate activity recognition models. The annotation process is prone to subjectivity, inconsistency, and errors on the part of human labelers. This can lead to unreliable or incorrect labels, which then degrade model performance. Strategies are needed to obtain higher quality and more consistent annotations. Another factor is synchronization between different sensor streams in multi-modal activity recognition. Systems combining data from multiple sensors like inertial measurement units on different body parts require precise synchronization between the sensor streams. Even small misalignments of a few milliseconds can significantly degrade model performance. Thus, robust synchronization techniques are necessary when working with multi-sensor setups. The Hawthorne effect refers to when participants change their behavior because they know they are being observed during an experiment. While psychologists have confirmed this effect, a data-driven user study was unable to reproduce it. Still, completely negating the potential impact of the Hawthorne effect on activity recognition datasets would be short-sighted. More research is needed to quantify if and how this observer effect manifests in different experimental protocols. Addressing these factors will enable more accurate deep learning for activity recognition. In addition, techniques like transfer learning and data augmentation can further enhance model performance. Transfer learning could leverage knowledge from pre-trained models, potentially improving accuracy and reducing training time. Data augmentation, by artificially expanding the dataset, can address potential biases and lead to more robust modeling, particularly for underrepresented classes.

Section 6.1

Conclusion

The contributions of this thesis can be categorized into four primary areas: **(1) The data recording system** - *Activate*, **(2) User Studies** that address research questions directly related to human activity recognition and deep learning, including the enhancement of recording methods, the synchronization of multiple sensors, and the assessment of annotation methods widely used in activity recognition, **(3) Deep Learning** aspects, such as *transfer learning* and *data augmentation*, and **(4)** a series of publications on **Sports Activity Recognition**, which interconnect all the aforementioned topics.

Data Recording System (1) and User Studies (2): The impact in this field is multifaceted. **I contributed an open-source recording system, *Activate*.** Its open-source character enables other researchers to adapt it according to their needs and reduces the hurdles to recording activity data for their projects.

The second contribution in this field of study highlights hurdles that arise while annotating data during in-the-wild studies. The quality of human activity recognition systems depends heavily on the underlying dataset annotations. However, manual annotations are inherently subjective and inconsistent. Human errors and variabilities during the annotation process propagate downstream, negatively impacting model performance. I conducted a study to quantify these annotation inconsistencies on real-world sensor-based activity data. My findings revealed significant inter-annotator disagreements, even between experts, when labeling the same activities using video ground truth. With the introduction of a data visualization and annotation tool were able to decrease the number of missing annotations and increase the annotation consistency. Therefore, the F1-Scores of the deep learning model have been increased by up to 8% (ranging between 82.1 and 90.4 % F1-Score). This highlights the difficulty of achieving consistent manual annotations. By quantifying annotation challenges and developing solutions to enhance consistency, this thesis enables the collection of more accurate and reliable benchmark datasets. **My contributions provide methodological improvements that will benefit future data collection efforts across the activity recognition community.**

Furthermore, I focused on the automated and precise synchronization of data recorded by multiple wearable sensors. This is critical for human activity analysis, yet non-trivial in practice. Drift between device clocks of ubiquitous systems gradually desynchronizes the sensor streams, even if correctly aligned initially. Uncontrolled sensor power-on sequences introduce further misalignments from the outset. To enable robust multi-sensor activity recognition, I developed an automated algorithm that accurately realigns drift-prone streams post-recording. This alleviates tedious manual synchronization while achieving higher precision than simplistic alignment assumptions. **My contribution provides a pragmatic solution for synchronizing data recorded by wearable devices**. I developed an algorithm to align multi-sensor

data by exploiting signal correlations. Cross-correlation detects matching patterns across the sensor streams indicating synchronization points. It then shifts the signals to align based on these anchored points. This avoids cumbersome manual alignment of long, continuous recordings. By applying this synchronization method I was able to synchronize independent data stream with a minimum mismatch of 0.30 seconds or 15 samples.

Another core contribution challenges the widely held view that the Hawthorne Effect causes people to modify behavior when monitored during observed and controlled studies. I conducted a data-driven study analyzing the Hawthorne Effect in activity recognition by having participants perform fitness activities in observed, semi-observed, and non-observed sessions. Participants wore a smartwatch running the Activate system to collect motion data. Through feature analysis and training a deep learning classifier on the multi-session data, I did not find significant differences between the observation conditions. While my study does not wholly refute the existence of the Hawthorne Effect, as evidenced by prior clinical trials, it does contest the common assumption about differences between laboratory and real-world data. Though some behavioral changes may occur when participants are observed, my results indicate classifiers can still learn similarly useful discriminative features.

By empirically investigating the Hawthorne Effect in an activity recognition context, my work provides new data-driven insights about the relevance of controlled studies for real-world applications. My findings challenge prevailing notions and will inform future protocol designs.

Deep Learning (3): I explored transfer learning and data augmentation techniques for improving model generalization in human activity recognition. Transfer learning by pre-training on large generic datasets can help, but performance gains are inconsistent across target tasks. **I propose tailored pre-training strategies more aligned with the activity recognition domain.** For data augmentation, I generated synthetic data using a generative adversarial network and evaluated samples with a DeepConv-LSTM. **My key contribution was developing novel strategies to select appropriate real samples to augment human activity datasets.** I investigated two selection strategies: participant-wise and fold-wise. The results prove that the success of the augmentation process heavily depends on the selected input data and the chosen augmentation strategy. Fold-wise augmentation increases the F1-Score by about 5.1% and subject-specific augmented data, increases the F1-Score by 11.0%.

Augmentation techniques are often copied from computer vision, where images can be flipped, rotated, cropped, or color-inverted. However, these transformations retain the core image information - an image of a dog remains an image of a dog. Though these techniques can work for HAR, directly applying them to sensor-based time-series data is illogical. Sensor data contains complex time-dependent information that standard augmentations would destroy, resulting in non-representative samples unrelated to the original class. Therefore, developing tailored augmentation techniques is essential for artificially increasing HAR training data. Standard image augmentations change

pixel values while retaining high-level content. However, altering sensor readings destroys precise time-series signatures needed for HAR models. Flipping, cropping, or rotating sensor data removes critical timing cues. While augmentations can still help tune model weights, generating realistic sensor data that preserves time-dependent relationships and class-discriminative features remains an open challenge. Effective data augmentation for HAR requires techniques specifically designed for sensor modalities, rather than directly adapting image-based approaches. Careful augmentation that considers sensor data complexity is needed to synthesize representative samples without distorting class-specific signatures. Despite progress in deep learning for HAR, focused research is still needed to uncover best practices surrounding transfer learning and data augmentation in this domain. Advancing knowledge in these areas will enable more rigorous leveraging of these techniques.

Sports Activity Recognition (4): I studied the capabilities and limitations of current deep learning architectures for recognizing the combination of periodic, complex, sporadic activities, using basketball as the domain of choice for a case study. Basketball provides a rich testbed with frequent changes in pace, direction, and intricate full-body movements. I trained state-of-the-art convolutional and recurrent neural networks on basketball activity recognition using inertial sensor data. The models were able to recognize the introduced basketball classes with limitations. Subject-independent F1-Scores were slightly above 50% for both architectures used. On the contrary, the overall performance for session-independent training shows a significant decrease and reaches around 24% for the shallow DeepConvLSTM [35] and 19% average F1-score for the Attend-and-Discriminate architecture [3]. **My analysis revealed remaining challenges in accurately detecting infrequent, unstructured motions like layups, rebounds, passes, or jumps-shots. The models struggled to differentiate subtle motion variations and lacked longer-term temporal context.** This highlights the need for more flexible networks with stronger reasoning capabilities.

In summary, this thesis has advanced the understanding of critical issues in activity recognition datasets - from the data collection process to annotation and analysis. The methods and insights provided here will enable the development of more accurate and robust activity recognition systems. Moving forward, I identify the recognition of complex, contextual activities as an open challenge for the field. As demonstrated through the basketball case study, sporadic activities require flexible models with greater reasoning capabilities. Transfer learning and data augmentation show promise for improving model generalization. Progress in this direction will expand the applicability of activity recognition to new real-world domains.

Section 6.2

Future Work

This Ph.D. thesis encompassed multiple topics related to human activity recognition. I believe further investment is warranted regarding annotation methodologies applied to human activity datasets. Robust and precise annotation methods especially for datasets recorded in-the-wild remain an underinvestigated area despite the importance of researchers developing new datasets. Weaknesses and strengths of certain methods must be discovered to overcome limitations. Results from such studies should be disseminated to the research community to develop methodologies that can be easily implemented and advanced by others. Our community still lacks powerful graphical annotation tools that surpass prototype status, undergo continuous improvement, and remain maintained. The study on the Hawthorne effect for wearable data recordings necessitates expanded participant cohorts, more diverse activities, and multiple recording environments to achieve sufficient representation across data collection scenarios potentially introducing bias.

The second critical aspect involved publishing the Hang-Time HAR dataset, which is the first dataset that is extensively focused on a single sport by targeting multiple basketball-specific activities. This enables researchers to concentrate on group activity recognition and activities with varying characteristics and challenges. For future work, I would like to augment this dataset with depth image data using a top-down camera below the basketball hoop and add context information of in-between players' activities.

Deep learning for human activity recognition presents numerous open research opportunities despite the maturity of some techniques. For example, transfer learning and data augmentation are established methods in deep learning, but unresolved questions remain surrounding their implementation for HAR. Regarding transfer learning for HAR, the field would benefit from determining optimal approaches such as sufficient data volume to enable model transfer, identifying data characteristics that allow transferability, preprocessing techniques that may be necessary to facilitate transfer, and which activity classes transfer well versus poorly. Although transfer learning shows promise for HAR, these open questions need to be addressed before it is widely adopted. Similarly, while data augmentation is an effective technique to increase training data volume, implementation details remain unclear in HAR applications. Potential research directions involve determining: the number of augmented training samples needed per activity class, which augmentation techniques work better for different HAR activity types, and which classes are difficult to augment realistically. Investigating these aspects would help the community establish best practices for effective data augmentation when working with HAR datasets.

List of Figures

List of Tables

List of Abbreviations

ADL Activities of Daily Living
AM.D. Arithmetic Mean of Dribble
API Application Programming Interface
ARM Advanced RISC Machines
ASCII American Standard Code for Information Interchange
BLE Bluetooth Low Energy
co crossover
CRS Comfort Rating Scale
CSV Comma-Separated Values
DeepConv-LSTM Deep Convolutional LSTM
ELAN EUDICO Linguistic Annotator
FFT Fast Fourier Transform
FIBA Fédération Internationale de Basketball
g Gravity of Earth
GAN Generative Adversarial Network
GMT Greenwich Mean Time
GNN Graph Neural Network
GPS Global Positioning System
GPU Graphics Processor Unit
GUI Graphical User Interface
HAR Human Activity Recognition
hd high dribbling
Hz Hertz
IMU Inertial Measurement Unit
IRB Institutional Review Board
js jump shot
kNN k-Nearest-Neighbor
ld low dribbling
LOSO Leave-One-Subject-Out
LSB Least Significant Bit
LSTM Long-Short-Term-Memory

MEMS Micro Electrical Mechanical System
MHz Megahertz
MoCap Motion Capturing
NBA National Basketball Association
NFC Near Field Communication
NLP Natural Language Processing
PAMAP2 Physical Activity Monitoring for Aging People
PCA Principle Component Analysis
PCB Printed Circuit Board
PPG Photoplethysmogram
ReLU Rectified Linear Unit
REST Representational State Transfer
RF Random Forest
RGB Red, Green, Blue
RISC Reduced Instruction Set Computer
RNN Recurrent Neural Network
sbj Subject
SEMMA Sample, Explore, Modify, Model, and Assess
SNR Signal-to-Noise-Ratio
SoC System-on-Chip
SQL Structured Query Language
SSL Secure Sockets Layer
TCN Temporal Convolutional Network
USA United States of America
WiFI Wireless Local Area Network

Bibliography

[1] *Deepl translator*, `https://www.deepl.com/translator`, Last accessed on 2023-11-07.

[2] *Grammarly*, `https://www.grammarly.com`, Last accessed on 2023-11-07.

[3] Alireza Abedin, Mahsa Ehsanpour, Qinfeng Shi, Hamid Rezatofighi, and Damith C. Ranasinghe, *Attend and discriminate: Beyond the state-of-the-art for human activity recognition using wearable sensors*, ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **5** (2021), no. 1, 1–22.

[4] Steven Abney, *Bootstrapping*, Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 360–367.

[5] U Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, and Hojjat Adeli, *Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals*, Computers in biology and medicine **100** (2018), 270–278.

[6] Rebecca Adaimi and Edison Thomaz, *Leveraging active learning and conditional mutual information to minimize data annotation in human activity recognition*, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **3** (2019), no. 3, 1–23.

[7] Adidas AG, *The future of football fits into your boot. smart tag, created in collaboration with ea sports fifa mobile and google*, 2021, `https://www.adidas.de/en/gmr_faq`,Last accessed on 2022-08-05.

[8] Preeti Agarwal and Mansaf Alam, *A lightweight deep learning model for human activity recognition on edge devices*, Procedia Computer Science **167** (2020), 2364–2373.

[9] Ali Akbari, Jonathan Martinez, and Roozbeh Jafari, *Facilitating human activity data annotation via context-aware change detection on smartwatches*, ACM Transactions on Embedded Computing Systems (TECS) **20** (2021), no. 2, 1–20.

[10] Omar AlShorman, Buthaynah AlShorman, Mahmood Alkhassaweneh, and Fahad Alkahtani, *A review of internet of medical things (iomt)–based remote health monitoring through wearable sensors: A case study for diabetic patients*, Indonesian Journal of Electrical Engineering and Computer Science **20** (2020), no. 1, 414–422.

[11] Kerem Altun, Billur Barshan, and Orkun Tunçel, *Comparative study on classifying human activities with miniature inertial and magnetic sensors*, Pattern

Recognition **43** (2010), no. 10, 3605–3620.

[12] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra Perez, and Jorge Luis Reyes Ortiz, *A public domain dataset for human activity recognition using smartphones*, Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning, 2013, pp. 437–442.

[13] Anthropic, *Claude*, `https://www.anthropic.com`, Last accessed on 2023-11-07.

[14] Marzieh M Ardestani and T George Hornby, *Effect of investigator observation on gait parameters in individuals with stroke*, Journal of biomechanics **100** (2020), 109602.

[15] Jacob S Arlotti, William O Carroll, Youness Afifi, Purva Talegaonkar, Luciano Albuquerque, John E Ball, Harish Chander, Adam Petway, et al., *Benefits of imu-based wearables in sports medicine: Narrative review*, International Journal of Kinesiology and Sports Science **10** (2022), no. 1, 36–43.

[16] Ron Artstein and Massimo Poesio, *Inter-coder agreement for computational linguistics*, Computational linguistics **34** (2008), no. 4, 555–596.

[17] Behrooz Azadi, Michael Haslgrübler, Bernhard Anzengruber-Tanase, Stefan Grünberger, and Alois Ferscha, *Alpine skiing activity recognition using smartphone's imus*, Sensors **22** (2022), no. 15, 5922.

[18] Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M Hausdorff, Nir Giladi, and Gerhard Troster, *Wearable assistant for parkinson's disease patients with the freezing of gait symptom*, IEEE Transactions on Information Technology in Biomedicine **14** (2009), no. 2, 436–446.

[19] Lu Bai, Christos Efstratiou, and Chee Siang Ang, *wesport: Utilising wrist-band sensing to detect player activities in basketball games*, 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), IEEE, 2016, pp. 1–6.

[20] Oresti Baños, Miguel Damas, Héctor Pomares, Ignacio Rojas, Máté Attila Tóth, and Oliver Amft, *A benchmark dataset to evaluate sensor displacement in activity recognition*, Proceedings of the 2012 ACM Conference on Ubiquitous Computing, 2012, pp. 1026–1035.

[21] Ling Bao and Stephen S Intille, *Activity recognition from user-annotated acceleration data*, International conference on pervasive computing, Springer, 2004, pp. 1–17.

[22] Rod Barrett, Maarten Vonk Noordegraaf, and Steven Morrison, *Gender differences in the variability of lower extremity kinematics during treadmill locomotion*, Journal of motor behavior **40** (2008), no. 1, 62–70.

[23] Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp, *Beat the ai: Investigating adversarial human annotation for reading comprehension*, Transactions of the Association for Computational Linguistics **8** (2020), 662–678.

[24] Bram JC Bastiaansen, Erik Wilmes, Michel S Brink, Cornelis J de Ruiter,

Geert JP Savelsbergh, Annemarijn Steijlen, Kaspar MB Jansen, Frans CT van der Helm, Edwin A Goedhart, Doris van der Laan, et al., *An inertial measurement unit based method to estimate hip and knee joint kinematics in team sport athletes on the field*, JoVE (Journal of Visualized Experiments) (2020), no. 159, e60857.

[25] Fabrizio Benedetti, Elisa Carlino, and Alessandro Piedimonte, *Increasing uncertainty in CNS clinical trials: the role of placebo, nocebo, and Hawthorne effects*, The Lancet Neurology **15** (2016), no. 7.

[26] Eugen Berlin and Kristof Van Laerhoven, *Detecting leisure activities with dense motif discovery*, Proceedings of the 2012 ACM Conference on Ubiquitous Computing, 2012, pp. 250–259.

[27] Eugen Berlin, Martin Zittel, Michael Braunlein, and Kristof Van Laerhoven, *Low-power lessons from designing a wearable logger for long-term deployments*, IEEE, apr 2015.

[28] Bison, Inc., *Product description of the basketball system hang-time*, 2023, `https://bisoninc.com/collections/hangtime`, Last accessed on 2023-06-23).

[29] J Martin Bland and DouglasG Altman, *Statistical methods for assessing agreement between two methods of clinical measurement*, The lancet **327** (1986), no. 8476, 307–310.

[30] Ulf Blanke, Diane Larlus, Kristof Van Laerhoven, and Bernt Schiele, *Standing on the shoulders of other researchers a position statement*, Pervasive Workshop, 2010.

[31] BM Sports Technology Gmbh, *The all-in-one solution for optimizing and controlling strength training*, 2022, `https://vmaxpro.de/`, Last accessed on 2022-08-05.

[32] Yang Bo, *A reinforcement learning-based basketball player activity recognition method using multisensors*, Mobile Information Systems **2022** (2022).

[33] Marius Bock, Alexander Hoelzemann, Michael Moeller, and Kristof Van Laerhoven, *Investigating (re) current state-of-the-art in human activity recognition datasets*, Frontiers in Computer Science **4** (2022), 924954.

[34] ———, *Investigating (re)current state-of-the-art in human activity recognition datasets*, Frontiers in Computer Science **4** (2022).

[35] Marius Bock, Alexander Hölzemann, Michael Moeller, and Kristof Van Laerhoven, *Improving deep learning for har with shallow lstms*, 2021 International Symposium on Wearable Computers, 2021, pp. 7–12.

[36] Marius Bock, Michael Moeller, Kristof Van Laerhoven, and Hilde Kuehne, *Wear: A multimodal dataset for wearable and egocentric video activity recognition*, arXiv preprint arXiv:2304.05088 (2023).

[37] Thiago O Borges, Alexandre Moreira, Renato Bacchi, Ronaldo L Finotti, Mayara Ramos, Charles R Lopes, and Marcelo S Aoki, *Validation of the vert wearable jump monitor device in elite youth volleyball players*, Biology of sport **34** (2017), no. 3, 239–242.

[38] Patrícia Bota, Joana Silva, Duarte Folgado, and Hugo Gamboa, *A semi-*

automatic annotation approach for human activity recognition, Sensors **19** (2019), no. 3, 501.

[39] Damien Brain and Geoffrey Webb, *On the effect of data set size on bias and variance in classification learning*, Proceedings of the Fourth Australian Knowledge Acquisition Workshop, University of New South Wales, 1999.

[40] Steven E Brenner, *Errors in genome annotation*, Trends in Genetics **15** (1999), no. 4, 132–133.

[41] Lorenzo Brognara, Antonio Mazzotti, Federica Rossi, Francesca Lamia, Elena Artioli, Cesare Faldini, and Francesco Traina, *Using wearable inertial sensors to monitor effectiveness of different types of customized orthoses during crossfit® training*, Sensors **23** (2023), no. 3, 1636.

[42] Niels P Brouwer, Ted Yeung, Maarten F Bobbert, and Thor F Besier, *3d trunk orientation measured using inertial measurement units during anatomical and dynamic sports motions*, Scandinavian Journal of Medicine & Science in Sports **31** (2021), no. 2, 358–370.

[43] Hennie Brugman, Albert Russel, and Xd Nijmegen, *Annotating multi-media/multi-modal resources with elan.*, LREC, 2004, pp. 2065–2068.

[44] Gino Brunner, Darya Melnyk, Birkir Sigfússon, and Roger Wattenhofer, *Swimming style recognition and lap counting using a smartwatch and deep learning*, Proceedings of the 23rd International Symposium on Wearable Computers, 2019, pp. 23–31.

[45] Andreas Bulling, Ulf Blanke, and Bernt Schiele, *A tutorial on human activity recognition using body-worn inertial sensors*, ACM Computing Surveys (CSUR) **46** (2014), no. 3, 1–33.

[46] Tyler Bushnell and Iain Hunter, *Differences in technique between sprinters and distance runners at equal and maximal speeds*, Sports biomechanics **6** (2007), no. 3, 261–268.

[47] Lauchlan Carey, Peter Stanwell, Douglas P Terry, Andrew S McIntosh, Shane V Caswell, Grant L Iverson, and Andrew J Gardner, *Verifying head impacts recorded by a wearable sensor using video footage in rugby league: a preliminary study*, Sports medicine-open **5** (2019), no. 1, 1–11.

[48] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz, *Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor*, 2015 IEEE International Conference on Image Processing (ICIP), IEEE, 2015, pp. 168–172.

[49] Hao Chen, Seung H. Cha, and Tae W. Kim, *A framework for group activity detection and recognition using smartphone sensors and beacons*, Building and Environment **158** (2019).

[50] Kaixuan Chen, Lina Yao, Dalin Zhang, Xianzhi Wang, Xiaojun Chang, and Feiping Nie, *A semisupervised recurrent convolutional attention model for human activity recognition*, IEEE transactions on neural networks and learning systems **31** (2019), no. 5, 1747–1756.

[51] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu, *Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities*, ACM Computing Surveys (CSUR) **54** (2021), no. 4, 1–40.

[52] Yiqiang Chen, Yang Gu, Xinlong Jiang, and Jindong Wang, *Ocean: A new opportunistic computing model for wearable activity recognition*, Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, 2016, pp. 33–36.

[53] Saisakul Chernbumroong, Anthony S Atkins, and Hongnian Yu, *Activity classification using a single wrist-worn accelerometer*, 2011 5th international conference on software, knowledge information, industrial management and applications (SKIMA) proceedings, IEEE, 2011, pp. 1–6.

[54] Jun-Ho Choi and Jong-Seok Lee, *Embracenet for activity: A deep multimodal fusion architecture for activity recognition*, Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, 2019, pp. 693–698.

[55] Mathias Ciliberto, Vitor Fortes Rey, Alberto Calatroni, Paul Lukowicz, and Daniel Roggen, *Opportunity ++: A multimodal dataset for video- and wearable, object and ambient sensors-based human activity recognition*, 2021.

[56] Ian Cleland, Manhyung Han, Chris Nugent, Hosung Lee, Sally McClean, Shuai Zhang, and Sungyoung Lee, *Evaluation of prompted annotation of activity data recorded from a smart phone*, Sensors **14** (2014), no. 9, 15861–15879.

[57] Dagoberto Cruz-Sandoval, Jessica Beltran-Marquez, Matias Garcia-Constantino, Luis A Gonzalez-Jasso, Jesus Favela, Irvin Hussein Lopez-Nava, Ian Cleland, Andrew Ennis, Netzahualcoyotl Hernandez-Cruz, Joseph Rafferty, et al., *Semi-automated data labeling for activity recognition in pervasive healthcare*, Sensors **19** (2019), no. 14, 3035.

[58] Alastair H Cummings, Mark S Nixon, and John N Carter, *A novel ray analogy for enrolment of ear biometrics*, 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), IEEE, 2010, pp. 1–6.

[59] Kimi D Dahl, Kristin M Dunford, Sarah A Wilson, Travis Lee Turnbull, and Scott Tashman, *Wearable sensor validation of sports-related movements for the lower extremity and trunk*, Medical Engineering & Physics **84** (2020), 144–150.

[60] Christophe De Vleeschouwer, Fan Chen, Damien Delannay, Christophe Parisot, Christophe Chaudy, Eric Martrou, Andrea Cavallaro, et al., *Distributed video acquisition and annotation for sport-event summarization*, NEM summit **8** (2008), no. 10.1016.

[61] Florenc Demrozi, Graziano Pravadelli, Azra Bihorac, and Parisa Rashidi, *Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey*, IEEE Access **8** (2020), 210816–210836.

[62] Terrance DeVries and Graham W Taylor, *Dataset augmentation in feature space*, arXiv preprint arXiv:1702.05538 (2017).

[63] Tom Diethe, Niall Twomey, and Peter A Flach, *Active transfer learning for activity recognition.*, ESANN, 2016.

[64] Anand Dubey, Niall Lyons, Avik Santra, and Ashutosh Pandey, *Xai-bayeshar: A novel framework for human activity recognition with integrated uncertainty and shapely values*, 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2022, pp. 1281–1288.

[65] Björn Eggert, Marion Mundt, and Bernd Markert, *Imu-based activity recognition of the basketball jump shot*, ISBS Proceedings Archive **38** (2020), no. 1, 344.

[66] Joseph Roland D Espiritu, *Aging-related sleep changes*, Clinics in geriatric medicine **24** (2008), no. 1, 1–14.

[67] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch, *Real-valued (medical) time series generation with recurrent conditional gans*, arXiv preprint arXiv:1706.02633 (2017).

[68] Simone Francia, Simone Calderara, and Dott Fabio Lanzi, *Classificazione di azioni cestistiche mediante tecniche di deep learning*, URL: https://www. researchgate. net/publication/330534530_Classificazione_di_Azioni_ Cestistiche_mediante_Tecniche_di_Deep_Learning (2018).

[69] Kenzie B Friesen, Zhaotong Zhang, Patrick G Monaghan, Gretchen D Oliver, and Jaimie A Roper, *All eyes on you: how researcher presence changes the way you walk*, Scientific Reports **10** (2020), no. 1, 1–8.

[70] Fédération Internationale de Basketball - FIBA, *Official basketball rules - fiba*, 2022, `https://www.fiba.basketball/basketball-rules`, Last accessed on 2022-08-09.

[71] Anna Lisa Gentile, Daniel Gruhl, Petar Ristoski, and Steve Welch, *Explore and exploit. dictionary expansion with human-in-the-loop*, The Semantic Web: 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2–6, 2019, Proceedings 16, Springer, 2019, pp. 131–145.

[72] Hassan Ghasemzadeh and Roozbeh Jafari, *Coordination analysis of human movements with body sensor networks: A signal processing model to evaluate baseball swings*, IEEE Sensors Journal **11** (2010), no. 3, 603–610.

[73] Hristijan Gjoreski, Mathias Ciliberto, Francisco Javier Ordoñez Morales, Daniel Roggen, Sami Mekki, and Stefan Valentin, *A versatile annotated dataset for multimodal locomotion analytics with mobile devices*, Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, 2017, pp. 1–2.

[74] Hristijan Gjoreski, Boštjan Kaluža, Matjaž Gams, Radoje Milić, and Mitja Luštrek, *Context-based ensemble method for human energy expenditure estimation*, Applied Soft Computing **37** (2015), 960–970.

[75] Martin Gjoreski, Vito Janko, Gašper Slapničar, Miha Mlakar, Nina Reščič, Jani Bizjak, Vid Drobnič, Matej Marinko, Nejc Mlakar, Mitja Luštrek, et al., *Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors*, Information Fusion (2020).

[76] Martin Gjoreski, Stefan Kalabakov, Mitja Luštrek, Matjaž Gams, and Hristijan

Gjoreski, *Cross-dataset deep transfer learning for activity recognition*, Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, 2019, pp. 714–718.

[77] Global Market Estimates, Inc., *Inertial measurement unit market insights*, 2022, `https://www.globalmarketestimates.com/market-report/inertial-measurement-unit-market-3331`, Last accessed on 2023-07-12.

[78] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, *Generative adversarial nets*, Advances in neural information processing systems, 2014, pp. 2672–2680.

[79] Google LLC, *Flutter.dev - an open source application framework*, 2022, `https://flutter.dev/`, Last accessed on 2022-08-12.

[80] Xiaofan Gu, Xinwei Xue, and Feng Wang, *Fine-grained action recognition on a novel basketball dataset*, ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 2563–2567.

[81] Mahdi Hamidi Rad, Vincent Gremeaux, Fabien Massé, Farzin Dadashi, and Kamiar Aminian, *Smartswim, a novel imu-based coaching assistance*, Sensors **22** (2022), no. 9, 3356.

[82] Nils Y. Hammerla, Shane Halloran, and Thomas Ploetz, *Deep, convolutional, and recurrent models for human activity recognition using wearables*, arXiv:1604.08880 [cs, stat] (2016).

[83] Nils Yannick Hammerla, James Fisher, Peter Andras, Lynn Rochester, Richard Walker, and Thomas Plötz, *Pd disease state assessment in naturalistic environments using deep learning*, Twenty-Ninth AAAI conference on artificial intelligence, 2015.

[84] Jason W Harding, Colin G Mackintosh, Allan G Hahn, and Daniel A James, *Classification of aerial acrobatics in elite half-pipe snowboarding using body mounted inertial sensors*, The Engineering of Sport **7** (2008), no. 2, 447–456.

[85] Ryosuke Hasegawa, Akira Uchiyama, and Teruo Higashino, *Maneuver classification in wheelchair basketball using inertial sensors*, 2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU), IEEE, 2019, pp. 1–6.

[86] Iqbal Hassan, Abtahi Mursalin, Robin Bin Salam, Nazmus Sakib, and HM Zabir Haque, *Autoact: An auto labeling approach based on activities of daily living in the wild domain*, 2021 Joint 10th International Conference on Informatics, Electronics & Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision & Pattern Recognition (icIVPR), IEEE, 2021, pp. 1–8.

[87] Sandro Hauri and Slobodan Vucetic, *Group activity recognition in basketball tracking data–neural embeddings in team sports (nets)*, arXiv preprint arXiv:2209.00451 (2022).

[88] Netzahualcoyotl Hernandez, Jens Lundström, Jesus Favela, Ian McChesney, and Bert Arnrich, *Literature review on transfer learning for human activity*

recognition using mobile and wearable devices with environmental technology, SN Computer Science **1** (2020), no. 2, 66.

[89] Alexander Hoelzeemann and Kristof Van Laerhoven, *Using wrist-worn activity recognition for basketball game analysis*, Proceedings of the 5th international Workshop on Sensor-based Activity Recognition and Interaction, 2018, pp. 1–6.

[90] Alexander Hoelzemann, Marius Bock, Ericka Andrea Valladares Bastías, Salma El Ouazzani Touhami, Kenza Nassiri, and Kristof Van Laerhoven, *A data-driven study on the hawthorne effect in sensor-based human activity recognition*, UbiComp/ISWC '23 Adjunct, Association for Computing Machinery, 2023, p. 486–491.

[91] Alexander Hoelzemann, Henry Odoemelem, and Kristof Van Laerhoven, *Using an in-ear wearable to annotate activity data across multiple inertial sensors*, Proceedings of the 1st International Workshop on Earable Computing, 2019, pp. 14–19.

[92] Alexander Hoelzemann, Jana Sabrina Pithan, and Kristof Van Laerhoven, *Open-source data collection for activity studies at scale*, Sensor-and Video-Based Activity and Behavior Computing, Springer, 2022, pp. 27–38.

[93] Alexander Hoelzemann, Julia Lee Romero, Marius Bock, Kristof Van Laerhoven, and Qin Lv, *Hang-time har: A benchmark dataset for basketball activity recognition using wrist-worn inertial sensors*, Sensors **23** (2023), no. 13, 5879.

[94] Alexander Hoelzemann, Nimish Sorathiya, and Kristof Van Laerhoven, *Data augmentation strategies for human activity data using generative adversarial neural networks*, 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), IEEE, 2021, pp. 8–13.

[95] Alexander Hoelzemann and Kristof Van Laerhoven, *Digging deeper: towards a better understanding of transfer learning for human activity recognition*, Proceedings of the 2020 International Symposium on Wearable Computers, 2020, pp. 50–54.

[96] ———, *A matter of annotation: An empirical study on in situ and self-recall activity annotations from wearable sensors*, arXiv preprint arXiv:2305.08752 (2023).

[97] Derek Hao Hu, Vincent Wenchen Zheng, and Qiang Yang, *Cross-domain activity recognition via transfer learning*, Pervasive and Mobile Computing **7** (2011), no. 3, 344–358.

[98] Iqram Hussain, Rafsan Jany, Richard Boyer, AKM Azad, Salem A Alyami, Se Jin Park, Md Mehedi Hasan, and Md Azam Hossain, *An explainable eeg-based human activity recognition model using machine-learning approach and lime*, Sensors **23** (2023), no. 17, 7452.

[99] Duy Tâm Gilles Huynh, *Human activity recognition with wearable sensors*, Technische Universität Darmstadt (2008), 59–65.

[100] Andrey Ignatov, *Real-time human activity recognition from accelerometer data*

using convolutional neural networks, Applied Soft Computing **62** (2018), 915–922.

[101] Hiroshi Inoue, *Data augmentation by pairing samples for images classification*, arXiv preprint arXiv:1801.02929 (2018).

[102] Sergey Ioffe and Christian Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, International conference on machine learning, PMLR, 2015, pp. 448–456.

[103] Diego Jaén-Carrillo, Luis E Roche-Seruendo, Alejandro Molina-Molina, Silvia Cardiel-Sánchez, Antonio Cartón-Llorente, and Felipe García-Pinillos, *Influence of the shod condition on running power output: An analysis in recreationally active endurance runners*, Sensors **22** (2022), no. 13, 4828.

[104] M. Jetté, K. Sidney, and G. Blümchen, *Metabolic equivalents (METS) in exercise testing, exercise prescription, and evaluation of functional capacity*, Clinical Cardiology **13** (1990), no. 8, 555–565.

[105] Fahim Kawsar, Chulhong Min, Akhil Mathur, and Alessandro Montanari, *Earables for personal-scale behavior analytics*, IEEE Pervasive Computing **17** (2018), no. 3, 83–89.

[106] Aftab Khan, James Nicholson, and Thomas Plötz, *Activity recognition for quality assessment of batting shots in cricket using a hierarchical representation*, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **1** (2017), no. 3, 1–31.

[107] Yasser Khan, Aminy E Ostfeld, Claire M Lochner, Adrien Pierre, and Ana C Arias, *Monitoring of vital signs with flexible and wearable medical devices*, Advanced Materials **28** (2016), no. 22, 4373–4395.

[108] Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych, *From zero to hero: Human-in-the-loop entity linking in low resource domains*, Proceedings of the 58th annual meeting of the Association for Computational Linguistics, 2020, pp. 6982–6993.

[109] J.F. Knight, C. Baber, A. Schwirtz, and H.W. Bristow, *The comfort assessment of wearable computers*, Sixth International Symposium on Wearable Computers (ISWC 2002),, IEEE Press, 2002.

[110] K. Kunze and P. Lukowicz, *Sensor placement variations in wearable activity recognition*, IEEE Pervasive Computing **13** (2014), no. 4, 32–41.

[111] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore, *Activity recognition using cell phone accelerometers*, ACM SigKDD Explorations Newsletter **12** (2011), no. 2, 74–82.

[112] Henry A. Landsberger, *Hawthorne revisited: Management and the worker, its critics, and developments in human relations in industry.*, Cornell Studies in Industrial and Labor Relations **9** (1958).

[113] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, *Deep learning*, Nature **521** (2015), no. 7553, 436–444.

[114] James B Lee, Rebecca B Mellifont, and Brendan J Burkett, *The use of a single inertial sensor to identify stride, step, and stance durations of running gait*,

Journal of Science and Medicine in Sport **13** (2010), no. 2, 270–273.

[115] SuKyoung Lee, Kyungsoo Kim, Yoon Hyuk Kim, and Seung-seob Lee, *Motion anlaysis in lower extremity joints during ski carving turns using wearble inertial sensors and plantar pressure sensors*, 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2017, pp. 695–698.

[116] Taylor Léger, Philippe J Renaud, Shawn M Robbins, and David J Pearsall, *Pilot study of embedded imu sensors and machine learning algorithms for automated ice hockey stick fitting*, Sensors **22** (2022), no. 9, 3419.

[117] Aleš Leonardis, Horst Bischof, and Jasna Maver, *Multiple eigenspaces*, Pattern recognition **35** (2002), no. 11, 2613–2627.

[118] Frédéric Li, Kimiaki Shirahama, Muhammad Adeel Nisar, Xinyu Huang, and Marcin Grzegorzek, *Deep transfer learning for time series data based on sensor modality classification*, Sensors **20** (2020), no. 15, 4271.

[119] Xiang Li, Daqing Zhang, Qin Lv, Jie Xiong, Shengjie Li, Yue Zhang, and Hong Mei, *Indotrack: Device-free indoor human tracking with commodity wi-fi*, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **1** (2017), no. 3, 1–22.

[120] Tianzheng Liao, Jinjin Zhao, Yushi Liu, Kamen Ivanov, Jing Xiong, and Yan Yan, *Deep transfer learning with graph neural network for sensor-based human activity recognition*, 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2022, pp. 2445–2452.

[121] Ding Liu, Ziyu Jin, and John Gambatese, *Scenarios for integrating ips–imu system with bim technology in construction safety control*, Practice Periodical on Structural Design and Construction **25** (2020), no. 1, 05019007.

[122] Hui Liu, Yale Hartmann, and Tanja Schultz, *Csl-share: A multimodal wearable sensor-based human activity dataset*, 2021.

[123] Li Liu, Yuxin Peng, Ming Liu, and Zigang Huang, *Sensor-based human activity recognition system with a multilayered model using time series shapelets*, Knowledge-Based Systems **90** (2015), 138–152.

[124] Li Liu, Yuxin Peng, Shu Wang, Ming Liu, and Zigang Huang, *Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors*, Information Sciences **340** (2016), 41–57.

[125] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan, *Learning transferable features with deep adaptation networks*, International conference on machine learning, PMLR, 2015, pp. 97–105.

[126] Yonggang Lu, Ye Wei, Li Liu, Jun Zhong, Letian Sun, and Ye Liu, *Towards unsupervised physical activity recognition using smartphone accelerometers*, Multimedia Tools and Applications **76** (2017), no. 8, 10701–10719.

[127] Mitja Luštrek and Boštjan Kaluža, *Fall detection and activity recognition with machine learning*, (2008).

[128] Chunyan Ma, Ji Fan, Jinghao Yao, and Tao Zhang, *Npu rgb+ d dataset and a feature-enhanced lstm-dgcn method for action recognition of basketball players*,

Applied Sciences **11** (2021), no. 10, 4426.

[129] Yuchao Ma, Andrew T Campbell, Diane J Cook, John Lach, Shwetak N Patel, Thomas Ploetz, Majid Sarrafzadeh, Donna Spruijt-Metz, and Hassan Ghasemzadeh, *Transfer learning for activity recognition in mobile health*, arXiv preprint arXiv:2007.06062 (2020).

[130] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong, *Bayesian loss for crowd count estimation with point supervision*, Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6142–6151.

[131] Kerry MacDonald, Roald Bahr, Jennifer Baltich, Jackie L Whittaker, and Willem H Meeuwisse, *Validation of an inertial measurement unit for the measurement of jump count and height*, Physical Therapy in Sport **25** (2017), 15–19.

[132] Andreas Madsen, *Running tensorflow lite on nodewatch/bangle.js - nearform*, Jul 2020.

[133] Andrii Maksai, Xinchao Wang, and Pascal Fua, *What players do with the ball: A physically constrained interaction modeling*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 972–981.

[134] Connor Malchow and Goeran Fiedler, *Effect of Observation on Lower Limb Prosthesis Gait Biomechanics: Preliminary Results*, Prosthetics and Orthotics International **40** (2016), no. 6.

[135] Marco Mangiarotti, Francesco Ferrise, Serena Graziosi, Francesco Tamburrino, and Monica Bordegoni, *A wearable device to detect in real-time bimanual gestures of basketball players during training sessions*, Journal of Computing and Information Science in Engineering **19** (2019), no. 1.

[136] Andrea Mannini, Stephen S Intille, Mary Rosenberger, Angelo M Sabatini, and William Haskell, *Activity recognition using a single accelerometer placed at the wrist or ankle*, Medicine and science in sports and exercise **45** (2013), no. 11, 2193.

[137] Christine F Martindale, Nils Roth, Julius Hannink, Sebastijan Sprager, and Bjoern M Eskofier, *Smart annotation tool for multi-sensor gait-based daily activity data*, 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE, 2018, pp. 549–554.

[138] Elton Mayo, *The human effect of mechanization*, The American Economic Review **20** (1930), no. 1.

[139] Jim McCambridge, John Witton, and Diana R. Elbourne, *Systematic Review of the Hawthorne Effect: New Concepts Are Needed to Study Research Participation Effects*, Journal of Clinical Epidemiology **67** (2014), no. 3.

[140] Sakorn Mekruksavanich and Anuchit Jitpattanakul, *Recognition of real-life activities with smartphone sensors using deep learning approaches*, 2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS), IEEE, 2021, pp. 243–246.

[141] Debasish Mishra, Rohan Basu Roy, Samik Dutta, Surjya K Pal, and Debashish Chakravarty, *A review on sensor based monitoring and control of friction stir*

welding process and a roadmap to industry 4.0, Journal of Manufacturing Processes **36** (2018), 373–397.

[142] Edmond Mitchell, David Monaghan, and Noel E O'Connor, *Classification of sporting activities using smartphone accelerometers*, Sensors **13** (2013), no. 4, 5317–5337.

[143] Tudor Miu, Paolo Missier, and Thomas Plötz, *Bootstrapping personalised human activity recognition models using online active learning*, 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, IEEE, 2015, pp. 1138–1147.

[144] Ghassem Mokhtari, Qing Zhang, Ghavameddin Nourbakhsh, Stephen Ball, and Mohanraj Karunanithi, *BLUESOUND: A new resident identification Sensor - Using ultrasound array and BLE technology for smart home platform*, IEEE Sensors Journal **17** (2017), no. 5.

[145] Riktim Mondal, Debadyuti Mukherjee, Pawan Kumar Singh, Vikrant Bhateja, and Ram Sarkar, *A new framework for smartphone sensor-based human activity recognition using graph neural network*, IEEE Sensors Journal **21** (2020), no. 10, 11461–11468.

[146] Francisco Javier Ordóñez Morales and Daniel Roggen, *Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations*, Proceedings of the 2016 ACM International Symposium on Wearable Computers, 2016, pp. 92–99.

[147] Carsten Müller, Christina Willberg, Lukas Reichert, and Karen Zentgraf, *External load analysis in beach handball using a local positioning system and inertial measurement units*, Sensors **22** (2022), no. 8, 3011.

[148] Chaithanya Kumar Mummadi, Frederic Philips Peter Leo, Keshav Deep Verma, Shivaji Kasireddy, Philipp M Scholl, Jochen Kempfle, and Kristof Van Laerhoven, *Real-time and embedded detection of hand gestures with an imu-based glove*, Informatics, vol. 5, MDPI, 2018, p. 28.

[149] Borja Muniz-Pardos, Shaun Sutehall, Jules Gellaerts, Mathieu Falbriard, Benoît Mariani, Andrew Bosch, Mersha Asrat, Jonathan Schaible, and Yannis P Pitsiladis, *Integration of wearable sensors into the evaluation of running economy and foot mechanics in elite runners*, Current sports medicine reports **17** (2018), no. 12, 480–488.

[150] Nitin Nair, Chinchu Thomas, and Dinesh Babu Jayagopi, *Human activity recognition using temporal convolutional network*, Proceedings of the 5th international Workshop on Sensor-based Activity Recognition and Interaction, 2018, pp. 1–8.

[151] Maryam Najafabadi, Flavio Villanustre, Taghi Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic, *Deep learning applications and challenges in big data analytics*, Journal of Big Data **2** (2015).

[152] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari, *Learning with noisy labels*, Advances in neural information processing systems **26** (2013).

[153] National Basketball Association - NBA, *Official basketball rules - nba*, 2022, `https://official.nba.com/`, Last accessed on 2022-08-09.

[154] David F Nettleton, Albert Orriols-Puig, and Albert Fornells, *A study of the effect of different types of noise on the precision of supervised learning techniques*, Artificial intelligence review **33** (2010), no. 4, 275–306.

[155] Le Nguyen Ngu Nguyen, Daniel Rodríguez-Martín, Andreu Català, Carlos Pérez-López, Albert Samà, and Andrea Cavallaro, *Basketball activity recognition using wearable inertial measurement units*, Proceedings of the XVI international conference on Human Computer Interaction, 2015, pp. 1–6.

[156] Lanshun Nie, Xue Li, Tianying Gong, and Dechen Zhan, *Few shot learning-based fast adaptation for human activity recognition*, Pattern Recognition Letters **159** (2022), 100–107.

[157] Malte Ollenschläger, Arne Küderle, Wolfgang Mehringer, Ann-Kristin Seifer, Jürgen Winkler, Heiko Gaßner, Felix Kluge, and Bjoern M Eskofier, *Mad gui: An open-source python package for annotation and analysis of time-series data*, Sensors **22** (2022), no. 15, 5849.

[158] OpenAI, *Chatgpt, v. 3.5*, 2023, `ttps://chat.openai.com/`, Last accessed on 2023-11-07.

[159] Francisco Javier Ordóñez and Daniel Roggen, *Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition*, Sensors **16** (2016), no. 1, 115.

[160] Iwona Pajak, Pascal Krutz, Justyna Patalas-Maliszewska, Matthias Rehm, Grezgorz Pajak, Holger Schlegel, and Martin Dix, *Sports activity recognition with uwb and inertial sensors using deep learning approach*, 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2022, pp. 1–8.

[161] Debajyoti Pal, Suree Funilkul, and Vajirasak Vanijja, *The future of smartwatches: assessing the end-users' continuous usage using an extended expectation-confirmation model*, Universal Access in the Information Society **19** (2020), 261–281.

[162] Luis Paredes, Ananya Ipsita, Juan C Mesa, Ramses V Martinez Garrido, and Karthik Ramani, *Stretchar: exploiting touch and stretch as a method of interaction for smart glasses using wearable straps*, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **6** (2022), no. 3, 1–26.

[163] Pascaline Parisot and Christophe De Vleeschouwer, *Scene-specific classifier for effective and efficient team sport players detection from a single calibrated camera*, Computer Vision and Image Understanding **159** (2017), 74–88.

[164] Aurélien Patoz, Thibault Lussiana, Bastiaan Breine, Cyrille Gindre, and Davide Malatesta, *A single sacral-mounted inertial measurement unit to estimate peak vertical ground reaction force, contact time, and flight time in running*, Sensors **22** (2022), no. 3, 784.

[165] Thomas Perri, Machar Reid, Alistair Murphy, Kieran Howle, and Rob Duffield, *Prototype machine learning algorithms from wearable technology to detect tennis*

stroke and movement actions, Sensors **22** (2022), no. 22, 8868.

[166] Aditya Ponnada, Seth Cooper, Binod Thapa-Chhetry, Josh Aaron Miller, Dinesh John, and Stephen Intille, *Designing videogames to crowdsource accelerometer data annotation for activity recognition research*, Proceedings of the Annual Symposium on Computer-Human Interaction in Play, 2019, pp. 135–147.

[167] Edward Purssell, Nicholas Drey, Jane Chudleigh, Sile Creedon, and Dinah J. Gould, *The Hawthorne effect on adherence to hand hygiene in patient care*, Journal of Hospital Infection **106** (2020), no. 2.

[168] Alen Rajšp and Iztok Fister, *A systematic literature review of intelligent data analysis methods for smart sport training*, Applied Sciences **10** (2020), no. 9, 3013.

[169] Vignesh Ramanathan, Jonathan Huang, Sami Abu-El-Haija, Alexander Gorban, Kevin Murphy, and Li Fei-Fei, *Detecting events and key actors in multi-person videos*, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3043–3053.

[170] Daniele Ravi, Charence Wong, Benny Lo, and Guang-Zhong Yang, *Deep learning for human activity recognition: A resource efficient implementation on low-power devices*, 2016 IEEE 13th international conference on wearable and implantable body sensor networks (BSN), IEEE, 2016, pp. 71–76.

[171] Christian Reich, Ahmad Mansour, and Kristof Van Laerhoven, *Embedding intelligent features for vibration-based machine condition monitoring*, 2018 26th European Signal Processing Conference (EUSIPCO), IEEE, 2018, pp. 371–375.

[172] Christopher Reining, Fernando Moya Rueda, Friedrich Niemann, Gernot A Fink, and Michael ten Hompel, *Annotation performance for multi-channel time series har dataset in logistics*, 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE, 2020, pp. 1–6.

[173] Attila Reiss and Didier Stricker, *Introducing a new benchmarked dataset for activity monitoring*, 2012 16th international symposium on wearable computers, IEEE, 2012, pp. 108–109.

[174] Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita, *Transition-aware human activity recognition using smartphones*, Neurocomputing **171** (2016), 754–767.

[175] Verónica Robles-García, Yoanna Corral-Bergantiños, Nelson Espinosa, María Amalia Jácome, Carlos García-Sancho, Javier Cudeiro, and Pablo Arias, *Spatiotemporal Gait Patterns During Overt and Covert Evaluation in Patients With Parkinson's Disease and Healthy Subjects: Is There a Hawthorne Effect?*, Journal of Applied Biomechanics **31** (2015), no. 3.

[176] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, et al., *Collecting complex activity datasets in highly rich networked sensor environments*, 2010 Seventh international conference on networked sensing systems (INSS), IEEE, 2010, pp. 233–240.

[177] ———, *Collecting complex activity datasets in highly rich networked sensor environments*, 2010 Seventh International Conference on Networked Sensing Systems (INSS), IEEE, 2010, pp. 233–240.

[178] Daniel Rojas-Valverde, Braulio Sánchez-Ureña, José Pino-Ortega, Carlos Gómez-Carmona, Randall Gutiérrez-Vargas, Rafael Timón, and Guillermo Olcina, *External workload indicators of muscle and kidney mechanical injury in endurance trail running*, International Journal of Environmental Research and Public Health **16** (2019), no. 20, 3909.

[179] Adrià Arbués Sangüesa, Thomas B Moeslund, Chris H Bahnsen, and Raul Benítez Iglesias, *Identifying basketball plays from sensor data; towards a low-cost automatic extraction of advanced statistics*, 2017 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE, 2017, pp. 894–901.

[180] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde, *Advanced interaction in context*, International Symposium on Handheld and Ubiquitous Computing, Springer, 1999.

[181] Philipp M Scholl, Matthias Wille, and Kristof Van Laerhoven, *Wearables in the wet lab: a laboratory system for capturing and guiding experiments*, Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2015, pp. 589–599.

[182] Max Schröder, Kristina Yordanova, Sebastian Bader, and Thomas Kirste, *Tool support for the online annotation of sensor data*, Proceedings of the 3rd International Workshop on Sensor-based Activity Recognition and Interaction, 2016, pp. 1–7.

[183] scikit-learn developers, *Cohen's kappa - scikit-learn*, 2022, `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score.html`, Last accessed on 2022-10-02.

[184] ———, *Principle component analysis - scikit-learn*, 2022, `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`, Last accessed on 2022-08-14.

[185] D Sculley, *Online active learning methods for fast label-efficient spam filtering.*, CEAS, vol. 7, 2007, p. 143.

[186] Umair Shafique and Haseeb Qaiser, *A comparative study of data mining process models (kdd, crisp-dm and semma)*, International Journal of Innovation and Scientific Research **12** (2014), no. 1, 217–222.

[187] Sarbagya Ratna Shakya, Chaoyang Zhang, and Zhaoxian Zhou, *Basketball-51: A video dataset for activity recognition in the basketball game*, CS & IT Conference Proceedings, vol. 11, CS & IT Conference Proceedings, 2021.

[188] Hoo-Chang Shin, Neil A Tenenholtz, Jameson K Rogers, Christopher G Schwarz, Matthew L Senjem, Jeffrey L Gunter, Katherine P Andriole, and Mark Michalski, *Medical image synthesis for data augmentation and anonymization using generative adversarial networks*, International workshop on simulation and synthesis in medical imaging, Springer, 2018, pp. 1–11.

[189] SIQ BASKETBALL, *Fiba approved smart ball*, 2020, `https://siqbasketball.com/`, Last accessed on 2022-08-05.

[190] Sky Deutschland Fernsehen GmbH & Co. KG, *Sky sports hang-time. 2023.*, 2023, `https://sport.sky.de/nba`, Last accessed on 2023-06-23).

[191] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee, *Learning from noisy labels with deep neural networks: A survey*, IEEE Transactions on Neural Networks and Learning Systems (2022).

[192] Konstantin Sozinov, Vladimir Vlassov, and Sarunas Girdzijauskas, *Human activity recognition using federated learning*, 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), IEEE, 2018, pp. 1103–1111.

[193] Craig A Staunton, Jonathan J Stanger, Daniel WT Wundersitz, Brett A Gordon, Edhem Custovic, and Michael IC Kingsley, *Criterion validity of a marg sensor to assess countermovement jump performance in elite basketballers*, The Journal of Strength & Conditioning Research **35** (2021), no. 3, 797–803.

[194] Maja Stikic, Diane Larlus, Sandra Ebert, and Bernt Schiele, *Weakly supervised recognition of daily life activities with wearable sensors*, IEEE transactions on pattern analysis and machine intelligence **33** (2011), no. 12, 2521–2537.

[195] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen, *Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition*, Proceedings of the 13th ACM conference on embedded networked sensor systems, 2015, pp. 127–140.

[196] Maike Stoeve, Dominik Schuldhaus, Axel Gamp, Constantin Zwick, and Bjoern M Eskofier, *From the laboratory to the field: Imu-based shot and pass detection in football training and game scenarios using deep learning*, Sensors **21** (2021), no. 9, 3071.

[197] Luka Svilar, Julen Castellano, Igor Jukic, and David Casamichana, *Positional differences in elite basketball: selecting appropriate training-load measures*, International journal of sports physiology and performance **13** (2018), no. 7, 947–952.

[198] Timo Sztyler and Heiner Stuckenschmidt, *On-Body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition*, IEEE International Conference on Pervasive Computing and Communications, 2016, pp. 1–9.

[199] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, Soren Brage, Nick Wareham, and Cecilia Mascolo, *Selfhar: Improving human activity recognition through self-training with unlabeled data*, Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies **5** (2021), no. 1, 1–30.

[200] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson, *Activity recognition in the home using simple and ubiquitous sensors*, International conference

on pervasive computing, Springer, 2004, pp. 158–175.

[201] Wolfgang Teufl, Markus Miezal, Bertram Taetz, Michael Fröhlich, and Gabriele Bleser, *Validity of inertial sensor based 3d joint kinematics of static and dynamic sport and physiotherapy specific movements*, PloS one **14** (2019), no. 2, e0213064.

[202] The Language Archive, MPI for Psycholinguistics Nijmegen, The Netherlands, *Elan-player timeseries viewer. 2023.*, 2023, `https://www.mpi.nl/corpus/html/elan/ch04s04s12.html`, Last accessed on 2023-01-23).

[203] The SciPy community, *Fast four transformation - the scipy community*, 2022, `https://docs.scipy.org/doc/scipy/tutorial/fft.html`, Last accessed on 2022-08-12.

[204] _____, *Local maxima - the scipy community*, 2022, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html`, Last accessed on 2022-08-12.

[205] _____, *Calculate the relative extrema of data*, 2023, `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.argrelextrema.html`, Last accessed on 2023-06-29.

[206] Changjia Tian, Varuna De Silva, Michael Caine, and Steve Swanson, *Use of machine learning to automate the identification of basketball strategies using whole team player tracking data*, Applied Sciences **10** (2019), no. 1, 24.

[207] Emma L Tonkin, Alison Burrows, Przemysław R Woznowski, Pawel Laskowski, Kristina Y Yordanova, Niall Twomey, and Ian J Craddock, *Talk, text, tag? understanding self-annotation of smart home data from a user's perspective*, Sensors **18** (2018), no. 7, 2365.

[208] Stewart G Trost, Yonglei Zheng, and Weng-Keen Wong, *Machine learning for activity recognition: hip versus wrist data*, Physiological measurement **35** (2014), no. 11, 2183.

[209] Ubiquitous Computing, University of Siegen, *Bangle.js connect app for android and ios. 2022.*, 2022, `https://github.com/ahoelzemann/Flutter_BangleJS_Connect`, Last accessed on 2023-06-20).

[210] _____, *Bangle.js connect app for android and ios. 2022.*, 2022, `https://ubi29.informatik.uni-siegen.de/upload/`, Last accessed on 2023-01-23).

[211] _____, *Custom firmware for the bangle.js. 2022.*, 2022, `https://github.com/kristofvl/BangleApps/tree/master/apps/activate`, Last accessed on 2023-06-20).

[212] Terry T. Um, Franz M. J. Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić, *Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks*, Proceedings of the 19th ACM International Conference on Multimodal Interaction (New York, NY, USA), ICMI '17, Association for Computing Machinery, 2017, p. 216–220.

[213] Yonatan Vaizman, Katherine Ellis, Gert Lanckriet, and Nadir Weibel, *Extrasensory app: Data collection in-the-wild with rich user interface to self-report*

behavior, Proceedings of the 2018 CHI conference on human factors in computing systems, 2018, pp. 1–12.

[214] Giulietta Valuri, Mark Stevenson, Caroline Finch, Peter Hamer, and Bruce Elliott, *The validity of a four week self-recall of sports injuries*, Injury Prevention **11** (2005), no. 3, 135–137.

[215] Kristof Van Laerhoven and Ozan Cakmakci, *What shall we teach our pants?*, Digest of Papers. Fourth International Symposium on Wearable Computers, IEEE, 2000.

[216] Kristof Van Laerhoven, Alexander Hoelzemann, Iris Pahmeier, Andrea Teti, and Lars Gabrys, *Validation of an open-source ambulatory assessment system in support of replicable activity studies*, German Journal of Exercise and Sport Research **52** (2022), no. 2, 262–272.

[217] Kristof Van Laerhoven, David Kilian, and Bernt Schiele, *Using rhythm awareness in long-term activity recognition*, 2008 12th IEEE International Symposium on Wearable Computers, IEEE, 2008, pp. 63–66.

[218] Shanmuga Venkatachalam, Harideep Nair, Ming Zeng, Cathy Shunwen Tan, Ole J Mengshoel, and John Paul Shen, *Semnet: Learning semantic attributes for human activity recognition with deep belief networks*, Frontiers in big Data (2022), 81.

[219] Joshua Vickers, Austin Reed, Robert Decker, Bryan P. Conrad, Marissa Olegario-Nebel, and Heather K. Vincent, *Effect of Investigator Observation on Gait Parameters in Individuals With and Without Chronic Low Back Pain*, Gait & Posture **53** (2017).

[220] Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber, *Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering*, Transactions of the Association for Computational Linguistics **7** (2019), 387–401.

[221] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu, *Deep learning for sensor-based activity recognition: A survey*, Pattern Recognition Letters **119** (2019), 3–11.

[222] Yufan Wang, Meng Chen, Xinyu Wang, Rosa HM Chan, and Wen J Li, *Iot for next-generation racket sports training*, IEEE Internet of Things Journal **5** (2018), no. 6, 4558–4566.

[223] Mark Weiser, *The computer for the 21st century*, Scientific american **265** (1991), no. 3, 94–105.

[224] David Whiteside, Olivia Cant, Molly Connolly, and Machar Reid, *Monitoring hitting load in tennis using inertial sensors and machine learning*, International journal of sports physiology and performance **12** (2017), no. 9, 1212–1217.

[225] Wikimedia Foundation, Inc., *Wikipedia article of the arcade game nba hangtime(1996)*, 2023, `https://en.wikipedia.org/wiki/NBA_Hangtime`, Last accessed on 2023-06-23).

[226] Gordon Williams, *The world's first open source hackable smart watch*, Bangle.js

- Hackable Smart Watch.

[227] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He, *A survey of human-in-the-loop for machine learning*, Future Generation Computer Systems (2022).

[228] Zhiwen Xiao, Xin Xu, Huanlai Xing, Fuhong Song, Xinhan Wang, and Bowen Zhao, *A federated learning system with enhanced feature extraction for human activity recognition*, Knowledge-Based Systems **229** (2021), 107338.

[229] Yan Yan, Dali Chen, Yushi Liu, Jinjin Zhao, Bo Wang, Xuankun Wu, Xiaohao Jiao, Yuqian Chen, Huihui Li, and Xuchao Ren, *Tnda-har*, 2021.

[230] Yue Yang, Li Wang, Steven Su, Mark Watsford, Lauren Marie Wood, and Rob Duffield, *Inertial sensor estimation of initial and terminal contact during in-field running*, Sensors **22** (2022), no. 13, 4812.

[231] Jie Yin, Qiang Yang, and Jeffrey Junfeng Pan, *Sensor-based abnormal human-activity detection*, IEEE Transactions on Knowledge and Data Engineering **20** (2008), no. 8, 1082–1090.

[232] Josh Jia-Ching Ying, Bo-Hau Lin, Vincent S. Tseng, and Sun-Yuan Hsieh, *Transfer learning on high variety domains for activity recognition*, Proceedings of the ASE BigData & SocialInformatics 2015 (New York, NY, USA), ASE BD&SI '15, Association for Computing Machinery, 2015, p. 6.

[233] Kristina Yordanova, *Challenges providing ground truth for pervasive healthcare systems*, IEEE Pervasive Computing **18** (2019), no. 2, 100–104.

[234] Kristina Y. Yordanova, Adeline Paiement, Max Schröder, Emma Tonkin, Przemyslaw Woznowski, Carl Magnus Olsson, Joseph Rafferty, and Timo Sztyler, *Challenges in annotation of user data for ubiquitous systems: Results from the 1st ARDUOUS workshop*, CoRR **abs/1803.05843** (2018).

[235] Chun Yu, Ting-Yuan Huang, and Hsi-Pin Ma, *Motion analysis of football kick based on an imu sensor*, Sensors **22** (2022), no. 16, 6244.

[236] Yisong Yue, Patrick Lucey, Peter Carr, Alina Bialkowski, and Iain Matthews, *Learning fine-grained spatial models for dynamic sports play prediction*, 2014 IEEE international conference on data mining, IEEE, 2014, pp. 670–679.

[237] Piero Zappi, Clemens Lombriser, Thomas Stiefmeier, Elisabetta Farella, Daniel Roggen, Luca Benini, and Gerhard Tröster, *Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection*, European Conference on Wireless Sensor Networks, Springer, 2008, pp. 17–33.

[238] Ming Zeng, Haoxiang Gao, Tong Yu, Ole J Mengshoel, Helge Langseth, Ian Lane, and Xiaobing Liu, *Understanding and improving recurrent networks for human activity recognition by continuous attention*, Proceedings of the 2018 ACM International Symposium on Wearable Computers, 2018, pp. 56–63.

[239] Bing Zhai, Yu Guan, Michael Catt, and Thomas Plötz, *Ubi-sleepnet: Advanced multimodal fusion techniques for three-stage sleep classification using ubiquitous sensing*, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **5** (2021), no. 4, 1–33.

[240] Shanshan Zhang, Lihong He, Eduard Dragut, and Slobodan Vucetic, *How to invest my time: Lessons from human-in-the-loop entity extraction*, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2305–2313.

[241] Kunlun Zhao, Junzhao Du, Congqi Li, Chunlong Zhang, Hui Liu, and Chi Xu, *Healthy: A diary system based on activity recognition using smartphone*, 2013 IEEE 10th international conference on mobile Ad-Hoc and sensor systems, IEEE, 2013, pp. 290–294.

[242] Lin Zhou, Eric Fischer, Can Tunca, Clemens Markus Brahms, Cem Ersoy, Urs Granacher, and Bert Arnrich, *How we found our imu: Guidelines to imu selection and a comparison of seven imus for pervasive healthcare applications*, Sensors **20** (2020), no. 15, 4090.

[243] Yexu Zhou, Haibin Zhao, Yiran Huang, Michael Hefenbrock, Till Riedel, and Michael Beigl, *Tinyhar: A lightweight deep learning model designed for human activity recognition*, International Symposium on Wearable Computers (ISWC'22), Atlanta, GA and Cambridge, UK, September 11-15, 2022, 2022.

[244] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, (2018).

[245] Tobias Zimmermann, Bertram Taetz, and Gabriele Bleser, *Imu-to-segment assignment and orientation alignment for the lower body using deep learning*, Sensors **18** (2018), no. 1, 302.