Smartwatch-based semantic learning of elements of human behaviour

Ada Alevizaki

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2023

In memory of my father, Lefteris, and godmother, Poppy.
To my friends and family.

Acknowledgements

Coming to the end of this PhD journey, I feel deeply fortunate to have met so many wonderful people who have shaped me on a professional and personal level into who I am today. To all of them, I devote this thesis.

I am deeply grateful to my supervisor, Prof. Niki Trigoni, for patiently and gently guiding me through my PhD thesis, and helping me find my research path all those times I could see no light at the end of the tunnel. Without her valuable understanding and support in academic and personal challenges, this thesis would not have been possible. I also sincerely thank my examiners, Prof. Ivan Martinovic and Dr. Juan Ye, for their valuable feedback, discussions and suggestions that significantly improved this thesis. Within the CPS group, I am indebted to Dr. Nhat (Nick) Pham, for his guidance and support over the past year, and to Dr. Stefano Rosa, for initial research discussions during the first year of my DPhil. Special thanks also to Pedro and Shuyu, for being incredible in and out of the office, and to Ben, for being the best office-mate I could ever have hoped for: sharing the stressful and happy moments of our lives, supporting and pushing each other, has really made office life special!

I am forever grateful to Wendy Poole, our amazing administrator in the AIMS CDT. Her organisational skills, her impeccable ability to having answers for any and all of my concerns, but also her caring and bubbly nature have made the Oxford experience feel safe, supported, and fun. True, and very big, thanks!

Dividing life between Oxford and London over the past few years, I am incredibly lucky to have been surrounded by so many people who each made life special in their own way. Big thanks to Yuki and Sid from the AIMS crew, for becoming close friends and sharing so many thoughts and moments (and coffees!). To Andreas, for starting as the best flatmate in Wolfson, for all the shared meals and long chats

in our living room over the lake, and for becoming the great friend he is today. To Kyriakos, Leo, Paula, Bernardo and Peny, for always making me feel at home in "450", for all the cooking, the dancing and the jokes, and for always finding the most lively places Oxford had to offer. To Artemis, Antonis, Marinos, Luli and Fani, for all the nights out, the Easter celebrations, for keeping the Greek vibe going, and for all their love, care and support. To Elpida, for all our long neighbourly chats that so gracefully juggled between very frivolous and deeply important, and for always being by my side. To Alessandro and Ornella, for the endless explorations around London's coolest neighbourhoods, our shared love for good food and coffee, the deep discussions, and their vivid presence in my life. To Ioanna and Notis, for becoming the great friends they are in such a short time. To Konstantinos, for remaining ever-present from far away in USA. To Dimitris, for the deep-dive into culture and philosophy, all the exhibitions and discussions about art and life. And to Nikitas, for having shared such a big part of this journey, and for all the caring and loving moments along the way.

I consider myself particularly fortunate for those people in my life who have managed to blur the lines between being friends and family. A big hug: to Hala, Reda, little Aïda and little Jude, for offering a home-away-from home, all our small and big shared moments; thanks, for being such an important part of my life! To Carolina and Veronica, for always keeping the mediterranean spirit alive, for so easily sharing our deepest thoughts, for motivating and supporting me and, really, for your love, shown in so many ways, from Oxford, to London, to Paris, to around Italy, and to wherever the future takes us. My biggest and warmest feelings are reserved for my life-long friends from Greece, Dimitra, Elpida and Eleni: I am extremely grateful for having you in my life, and am so proud of how we navigate life alongside each other - loving, encouraging, caring for, and helping each other grow, I can't wait to see what life holds for us down the road!

No matter how far we go, we always carry family inside us. To my father Lefteris and my godmother Poppy, who only saw the beginning of this journey: I am grateful that you believed in me, for all your love, and your excitement in what was gradually becoming my new life in the UK. To my cousins Kostis and Dimitra, and our sweet Ioli and Xenia: I am incredibly lucky for having you be an active part of my life in Oxford, for all the warm hugs, the giggles and the playful moments, and for all your love and care. Finally, to my mother Angeliki, my aunts and uncles Ioanna and Vasilis, Maria and Ilpo, and my dear grandmother, Georgia: thanks, for everything, always. You are eternally a part of me.

Abstract

The modern interconnected world is characterised by a plethora of mobile applications that operate in smart mobile and wearable devices, aiming to assist various aspects of human life, such as smart home operations or remote monitoring of older adults. Understanding daily patterns of human behaviour is key to providing informed and robust solutions in such scenarios. By inspecting the focus of popular mobile applications closely it becomes apparent that they commonly share three main requirements: accurate location estimates; robust activity identification; and user perception of the task performed.

While research in the area of ubiquitous computing has provided accurate large-scale solutions for these problems, the design specifications of mobile applications that enhance a user's everyday life call for solutions that allow for a consistent flow of information, are cost-efficient and non-disruptive to the user's natural way of conduct, and preserve the privacy of the user and of others who share the environment. While smartphone-based approaches would satisfy most of the above requirements, users do not carry the device with them at all times, particularly in private indoor spaces such as their home or workplace, disrupting the continuity of tracking. Even when they do, the common placement of the device in the user's pocket or bag restricts the range of activities that can be detected, as manual or tactile activities, for example, cannot be captured.

This thesis aims to overcome these challenges by addressing the outlined problems from a semantic perspective using a smartwatch as the main sensor modality. In particular, we attempt to answer the following fundamental question: *can we establish a seamless, continuous, and centralised semantic understanding of a human's location and activity content, and use these concepts to build robust models that accurately capture and explain naturalistic human behaviours?*

Smartwatches are lightweight and widely used in modern societies, and their premium wrist attachment allows for a variety of activities to be recorded and for the continuity of data information. At the same time, semantic rather than precise solutions enhance user privacy, as they relieve the need for environment maps or estimation of a user's exact pose when performing an activity.

As smartwatch-based solutions have not been widely explored in the past, there is a lack of publically available datasets of smartwatch sensor recordings, particularly in the area of human positioning. In this thesis, we first attempt to address this challenge by contributing two relevant datasets: **WatchBLoc**, a dataset of smartwatch inertial sensor data and ambient BLE signals, coupled with ground truth location information; and **watchHAR**, an activity dataset of smartwatch inertial sensor data, enhanced with accurate geographical location data from the VICON infrared positioning system.

Second, we address the indoor positioning problem in private indoor spaces. Assuming most modern household devices are equipped with Bluetooth IoT sensors and that the user owns and consistently wears a smartwatch, we propose **RoomE**, a probabilistic algorithm that achieves semantic localisation and room identification by sequentially fusing ambient location information with high-level mobility features derived from inertial data, no matter the IoT devices' placement in the environment.

Third, we study the problem of human activity recognition from a centralised perspective. In this work, we identify the vast number of mobile applications that operate on smartwatches often present overlapping requirements of behavioural information. Motivated by the restricted computing resources of smartwatches, we propose **CHiHAR**, a centralised hierarchical activity recognition framework that expresses activities at different levels of abstraction over a hierarchy of semantic levels. This allows for a centralised pool of information, from which each mobile application can source the necessary activity information, at its required level of detail, only when required.

Finally, we demonstrate the challenges that arise in the explainability of datasets that have been collected in uncontrolled conditions, such as crowdsourced datasets, even when utilising models trained on semantically similar datasets. We identify four levels of semantic challenges, relating to human task perception and annotation consistency, that contribute to this difficulty and hinder the generalisation and knowledge transfer capabilities of learned models. To mitigate these, we demonstrate **SeMEDA**, a structured approach to bridge semantic inconsistencies between datasets and identify the underlying rules that govern the activities encoded in an incoming, challenging dataset.

All the proposed approaches are comprehensively evaluated through extensive experiments, and the results demonstrate their potential impact on a broad spectrum of human behaviours.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Advances in the area of ubiquitous computing have paved the way for a plethora of mobile applications operating in smart devices that aim to assist various aspects of everyday life. From human-environment interaction [1, 2] to remote monitoring [3, 4] or general improvement of personal goals, smart devices have become a common part of daily life. Smartphone use is almost universal nowadays [5], while other connected devices, such as smartwatches [6], smart rings or ear wearables [7, 8], are becoming increasingly popular.

The rising human interaction with mobile devices [11] and their inherent capabilities to record user data via their large number of embedded sensors provides an exciting opportunity for a mutually beneficial exchange of information between users and mobile applications: while users can get useful insight into various aspects of everyday life, mobile applications can also further enhance their performance, simply by having access to the data logged in the devices when the users interact with, or carry them.

As mobile and wearable devices become more widely available and socially acceptable, and their computational power increases, there is also an expansion to

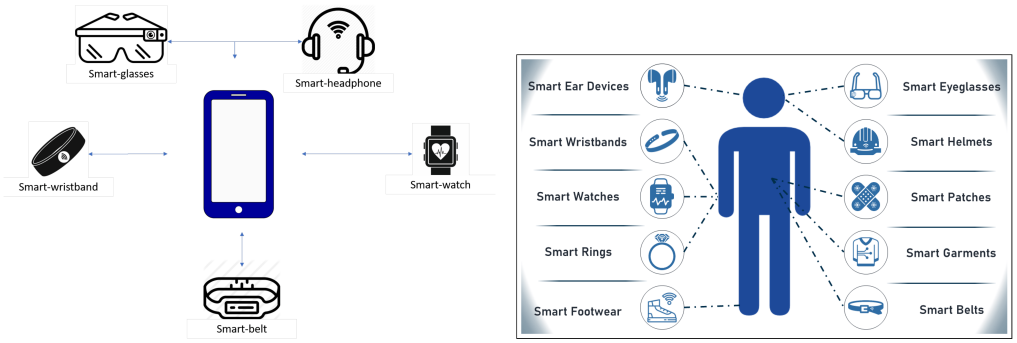


Figure 1.1: Example of smart wearable devices [9, 10].

Smart wearable devices are becoming increasingly popular. With smartphone usage almost universal, several other interconnected smart wearables are becoming accessible to the average user, encouraging human-device interactions.

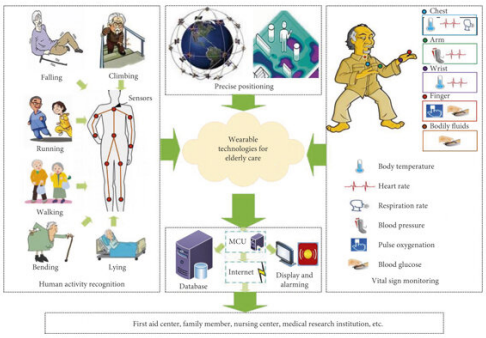
the fields where mobile applications targeted to human behaviour understanding can have a significant impact. Fig. 1.2 provides some illustrative examples: fitness trackers, in the form of step counting or identifying the form of exercise (e.g., walking, cycling, etc.) are some of the most popular applications nowadays [12]; outdoor navigation and mapping solutions (e.g., Strava, Beeline) are practically seamless due to GPS technology [13]; indoor positioning and navigation for public indoor spaces, that are equipped with multiple Wi-Fi Access Points (APs) or Bluetooth Low Energy (BLE) beacons, has also seen significant advancement in the last few years, such as customer navigation in shopping centres [14] and workflow optimisation in hospitals [15]; the wearable health market is also seeing increasing popularity, both in the context of diagnosis and monitoring of disease-progress or rehabilitation-process [16, 17] and remote monitoring and assistance of older adults in face of the "effect of the ageing population" [18]; another field on the rise is smart home automation [19], allowing for houses to regulate temperature, lighting, etc., based on the inhabitants' usage patterns; and many more.

By examining the above scenarios closely, it becomes apparent that there are three main aspects governing the understanding of human behaviours: ***where*** people are (***human localisation***), ***what*** they do (***human activity recognition***), and ***how*** they do it (***human task perception***). An important detail in addressing

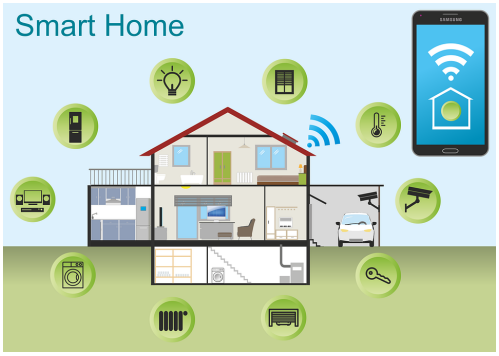
(a) Fitness trackers [20].

(b) GPS navigation [21].

(c) Indoor positioning [22].



(d) Remote monitoring [23].



(e) Smart homes [24].

Figure 1.2: Mobile applications for human behaviour understanding.

Examples of mobile applications related to human behavioural patterns. Data are recorded in embedded sensors of wearable devices, such as phones, smartphones, smartwatches, etc., in conjunction with ambient sensors when necessary. (a) Fitness trackers, e.g., pedometers, activity trackers, etc. (b) GPS positioning and navigation in outdoor spaces. (c) Indoor positioning in GPS-denied environments. (d) Remote monitoring of older adults. (e) Smart home management.

this problem pertains to the nature of mobile applications, which in the modern interconnected world strive to support and enhance the user's daily activities. Consequently, it is pivotal for a system to effectively extract this information seamlessly without expecting the user to alter their natural conduct or use extensive equipment, or imposing heavy infrastructure on the user's environment. This challenge serves as the central overarching theme of this dissertation.

1.2 Research scope

As we delve into the subsequent chapters, we advocate that a comprehensive solution to the stated problem of human behavioural understanding can be established through a *semantic interpretation* of the above concepts, using a simple *smartwatch, wrist-attached wearable*, complemented solely by ambient sensors, when required.

Despite significant research contributions in the areas of human positioning and activity recognition in the last few years, and an increased tendency towards mobile solutions, smartwatch-based systems have not been thoroughly explored in either of these research directions. As a result, there is a lack of available datasets recorded by smartwatch sensors, particularly in the area of indoor positioning. With the problem of outdoor positioning solved by GPS technology, most current research is focused on Indoor Positioning Systems (IPS). In *public* indoor spaces, various methods have been developed, using cameras [25], LiDAR sensors [26], mmWave sensors [27], infrared (IF) systems [28] and, more commonly, a combination of RF-based localisation and inertial data from smartphones [29–31]. Similarly to the positioning scenario, the HAR problem has been addressed widely by the machine learning community, with a variety of sensor networks, such as cameras [32, 33], RFID sensors [34], mmWave [35, 36] and wearable sensors, with smartphones being the primary sensing modality [37–39], due to their widespread usage, long battery life and computing power. While smartwatches may lack some of these characteristics, they also present unique opportunities. In Fig. 1.3, we provide an illustrative comparison between smartphones and smartwatches, showcasing the advantages and challenges of each chosen modality.

The premium wrist-attached location of a smartwatch allows for secure, continuous attachment of the device to the user, allowing for uninterrupted monitoring. The increased popularity and social acceptability of smart wearables also makes the device non-intrusive to the user, allowing for natural conduct. In the area of Human Activity Recognition (HAR) in particular, smartwatches and other wearables also offer the potential to broaden the range of activities detectable by a HAR system. While a smartphone may effectively capture simple activities like walking

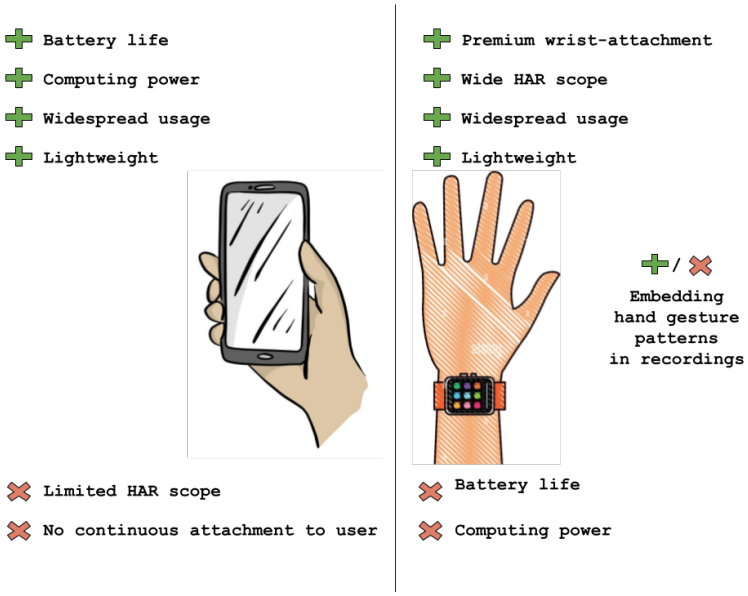


Figure 1.3: Challenges and advantages of smartphone and smartwatch sensors in indoor positioning and HAR.

Smartphones have become the most popular mobile sensor in recent advances in indoor positioning and activity recognition, due to their widespread usage, small size and sufficient computing power and battery life. However, their common placement in the user's bags or pockets limits the scope of activities to be identified. At the same time, it is required for the user to carry the device at all times for effective monitoring, which is not the case in private indoor environments. While smartwatches may fall short in their computational capacity, they are also becoming increasingly popular in modern societies. Their wrist-attachment ensures continuous data flow, and allows for an increased scope of activities to be performed, extending to all activities that involve manual dexterity. As hands are involved in virtually every activity a human performs, incurring noise and making it challenging to record pure activities where hand usage is only complementary, this characteristic also allows the sensors to capture more naturalistic patterns in human behaviour, where people commonly involve hand movements to many other primary activities.

or sitting, it may fall short in capturing activities requiring manual dexterity, such as brushing teeth. In such cases, a smartwatch could prove beneficial, while a smart ring might extend the system's predictive capabilities even further, to include tactile activities like typing, drawing or sewing.

The distributed capacity of a multi-sensor wearable system to identify a diverse array of activities gains further significance when considering the requirements of various mobile applications commonly used by the average user. Despite serving distinct purposes, many applications rely on similar information about the user's

activity patterns. For instance, a remote monitoring application could enhance its positioning estimates by leveraging information about the user brushing their teeth, indicating their likely presence in the bathroom. Similarly, a health tracker might also require this information to monitor dental hygiene. Importantly, the exact semantic information need not be identical in both cases: while precise details about the action of "brushing teeth" are crucial for dental hygiene monitoring, a remote monitoring application could be equally informed by identifying a more generic "activity commonly performed in the bathroom". This example illustrates that identifying activity patterns in a centralised manner and at various levels of hierarchical abstraction can significantly benefit multiple mobile applications simultaneously, optimising computational resources by making information available to multiple applications at once. While some hierarchical approaches to activity recognition have been explored [40, 41], this has not been addressed in a systematic way.

The increased popularity of smartwatches and other wearable and mobile devices equipped with various mobile applications also allow for the collection of large crowdsourced datasets, collected and self annotated by several users in real-world conditions. Crowdsourced datasets are valuable to the research community, as they capture realistic patterns of human behaviour. At the same time, however, learning from, or transferring knowledge to crowdsourced datasets is particularly challenging. Annotations are often inconsistent, missing or incorrect, as there is usually limited annotation guidance; labels are assigned based on the personal understanding and attention to detail of each individual user. While the research area of transfer learning has explored domain adaptation techniques between different sensor modalities, sensor placements or users [42], there is no particular focus to understanding the semantic intricacies of such datasets. Techniques relating to noisy label handling may seem like an appropriate direction; nonetheless, crowdsourced datasets often exhibit significant variability in the perception of class definitions among users, inhibiting robust learning under such approaches. Smartwatch-based crowdsourced data, that are rich in their recorded activity patterns, present a unique

opportunity for utilising generalisation and transfer learning approaches from a controlled dataset to explain a crowdsourced dataset's underlying patterns.

Based on the above discussion, this thesis focuses on the use of a smartwatch as the sole sensing modality to tackle the problems of human indoor localisation, activity recognition and activity data explainability from a semantic perspective, offering seamless, continuous and centralised solutions. In particular, we will explore:

- The lack of publicly available location and activity data recorded by smartwatches due to their limited use in research to date, which hinders consistent benchmarking across relevant research.
- The capabilities of a single smartwatch to estimate and continuously track the room-level location of the bearer of the device using ambient sensor signals without additional infrastructure.
- How smart devices are capable of not only identifying the activities performed by the bearer of the device but also sharing this information in various hierarchical levels of detail with the multiple mobile applications installed on the device which may require the same information from different perspectives.
- How naturalistic human behaviour, albeit valuable, contributes to inconsistent datasets with noisy labels. Acknowledging this, we will explore how a semantic hierarchy defined over a controlled dataset can help alleviate such data discrepancies, revealing relevant underlying semantic concepts in any dataset with semantic proximity to the training data.

Considering the goals summarised above, the main question of this thesis is: *can we establish a seamless, continuous, and centralised semantic understanding of a human's location and activity content, and use these concepts to build robust models that accurately capture and explain naturalistic human behaviours?*

In the next section, we will highlight the challenges for indoor positioning, HAR and activity data explainability in the context of continuous, non-intrusive monitoring of users, as well as the additional challenges that arise from our proposed smartwatch-based semantic learning approach.

1.3 Research challenges

In the previous section, we established that understanding naturalistic human behavioural patterns involves non-intrusive, continuous and centralised understanding of the person's location and activity. This research aims to overcome real issues faced in complex real-world environments to achieve robust and ubiquitous semantic human localisation and activity recognition. We proposed addressing these using a wrist-attached smartwatch and ambient sensors. In this section, we highlight the high-level challenges for each of the above problems: indoor positioning, human activity recognition, and activity data inconsistencies, due to varying human perception of a task and annotation errors during data collection.

1.3.1 Data availability challenges

While smartphone inertial sensors are popular choices in both indoor positioning and HAR research, smartwatch sensors are significantly underexplored. Though this is reflected primarily in the existing methodologies in indoor positioning and HAR, it also extends to the lack of available datasets recorded with smartwatch sensors.

This is particularly relevant to the area of indoor positioning, where we have identified only one relevant dataset containing smartwatch recordings [43].

In the area of HAR, a few studies include inertial smartwatch recordings alongside other sensor modalities, such as smartphones [44–47]. However, the majority of these include a limited set of activities [46], for example walking, sitting and running, are focused on low-level actions [44], or suffer from significant inconsistencies and label noise, as they were crowdsourced [47].

While naturalistic datasets are valuable in capturing realistic human behaviours, it is also essential to offer well-controlled datasets that allow for robust and precise model training.

1.3.2 Indoor positioning challenges

Our goal regarding human indoor positioning is to establish accurate semantic location of a user avoiding heavy infrastructure and ensuring continuous tracking. We identify the following challenges:

- Existing indoor positioning methods are cumbersome and expensive, or very intrusive to the user's space and behaviour.
- Furthermore, it is difficult for methods to generalise across different spaces, as they often require the use of floorplans, extensive fingerprinting or crowd-sourcing.
- In particular in private indoor spaces, such as houses, distances between rooms are constrained, allowing ambient signals (WiFi or BLE) to easily propagate across nearby spaces. As such, they often create similar fingerprints so it is difficult to distinguish adjacent rooms, especially compared to large spaces.
- The placement of RF-based ambient sensors can significantly affect the overall accuracy, however, it has not been studied extensively in literature.

To overcome these challenges, we propose in Chapter 4.7 an approach that utilises only inertial data from a wrist-attached smartwatch, and ambient bluetooth sensors. While this approach ensures continuous user tracking with minimal user disturbance, it also comes with its unique set of challenges.

- While sensor quality and battery power are continually improving in wearable sensors, such as smartwatches, these remain significantly less powerful compared to smartphone sensors. To this end, scanning frequently for ambient sensor signals can take a toll on the device's battery life, and limited scanning can affect the continuity of positioning estimates. Finding an informative yet battery power-preserving scanning frequency is an important practical problem in human indoor positioning.

1.3.3 Human activity recognition challenges

In this thesis, we view human activity recognition at various levels of abstraction, where the same activity can acquire multiple labels of different descriptive detail (e.g., coarse or detailed). We identify the following challenges:

- Many successful existing approaches require cumbersome infrastructure (e.g., mmWave, LiDAR) or can raise privacy concerns for the monitored users (e.g., video-based HAR).
- Smartphones as an alternative overcome these issues, but are impractical for continuous HAR, as they are not always carried by users in private indoor spaces.
- Smartphones also have a limited taxonomy of activities, as they cannot effectively capture activities that entail more intricate movements, such as manual or tactile events.

To overcome these challenges, we propose in Chapter 5.7 a hierarchical HAR approach that utilises only inertial data from a wrist-attached smartwatch. While this approach ensures continuous user tracking with minimal user disturbance, and allows for centralised predictions at various levels of abstraction, it raises the following additional challenges:

- Though smartwatches can achieve continuous monitoring due to their wrist-attachment, they have limited computational resources. As such, efficiently managing the demands of numerous mobile applications that often request the same information (albeit at different timings or levels of detail) is a key challenge in smartwatch-based HAR.
- The hand-involvement on most activities in smartwatch data collection leads to similar signal patterns for distinct activities within a dataset, significantly hindering a system's capacity to learn robust, accurate and generalisable models on pure activities.

- Defining an activity at various levels of abstraction alleviates the above challenges. However, identifying the hierarchical abstraction over a set of activities is not trivial; this is further hindered by the naturalistic behaviour encoded in smartwatch activity data, due to the engaging hand movements.

1.3.4 Data explainability challenges

The notion of data explainability in this thesis is viewed through the scope of generalisation and knowledge transfer to a similar but not semantically identical target domain. Suppose two HAR datasets defined over semantically similar activity classes; one of the datasets is considered carefully curated and well-annotated (source), while the second dataset is crowdsourced (target). We identify the following challenges in generalising, or transferring knowledge from the source to the target dataset:

- The two datasets may contain different classes, albeit semantically similar. In these cases, standard out-of-distribution detection for generalisation is not sufficient.
- Annotations may be assigned over varying time windows, leading to distinct feature and output spaces across the two datasets.
- The underlying perception between two respective classes of the same (or equivalent, e.g., walking - WALK) name in the two datasets may be different, incurring label inconsistency among the two datasets.
- The underlying perception of a class among users of the crowdsourced dataset may be variable, leading to within-class inconsistencies.

To overcome these challenges, we propose in Chapter 6.7 an inverse-hierarchy approach to semantic mapping across such datasets. While the choice of smartwatch is not absolutely necessary and the method could be applied to data from any wearable device, smartwatch activity datasets tend to be more appropriate, as they usually involve a wider range of activities, hence it is more straightforward to define a robust hierarchy at different levels of abstraction.

In this section, we have highlighted the challenges that arise when attempting to address the problem of continuous, non-intrusive monitoring of human behavioural patterns, giving particular focus on the challenges that arise from the selected smartwatch sensor modality. In the next section, we will summarise the contributions of this thesis towards this goal.

1.4 Research contributions

In summary, we make the following contributions in this thesis:

- Two smartwatch-based datasets to improve the benchmark availability on smartwatch data in the research community.
  - The first comprehensive dataset, **WatchBLoc**, comprising signals from BLE home devices and user-worn smartwatches allowing for the study of room-level localisation under a variety of conditions: users, IoT device layouts and home layouts. This contribution is described in the following publication:
    - **Alevizaki, A.**, Trigoni, N. (2022). RoomE and WatchBLoc: A BLE and Smartwatch IMU Algorithm and Dataset for Room-Level Localisation in Privacy-Preserving Environments. In the 19th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2022).
    - The dataset is publicly available at:

      `https://doi.org/10.5281/zenodo.7039554`.
  - A smartwatch IMU dataset for activities of daily living, **watchHAR**. This contribution is described in the following publication:
    - **Alevizaki, A.**, Pham, N., Trigoni, N. (2023). Hierarchical Activity Recognition with Smartwatch IMU. In the 24th International Conference on Distributed Computing and Networking (ICDCN '23).

∗ The dataset is publicly available at:

`https://doi.org/10.5281/zenodo.7092552`.

- A probabilistic algorithm that achieves semantic localisation and room identification by sequentially fusing ambient location information with high-level mobility features derived from inertial data, no matter the IoT devices' placement in the environment. It also provides the capability of inferring room transition probabilities without using floorplan information. This contribution is described in the following publication:

  – **Alevizaki, A.**, Trigoni, N. (2022). RoomE and WatchBLoc: A BLE and Smartwatch IMU Algorithm and Dataset for Room-Level Localisation in Privacy-Preserving Environments. In the 19th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2022).

- A hierarchical framework that achieves robust classification performance for various ADLs, such as walking, personal hygiene and household tasks, at a low computational cost over a hierarchy of decision levels. Our method is easily deployed as it only requires a wrist-attached smartwatch without any requirements for calibration or user constraints. This contribution is described in the following publication:

  – **Alevizaki, A.**, Pham, N., Trigoni, N. (2023). Hierarchical Activity Recognition with Smartwatch IMU. In the 24th International Conference on Distributed Computing and Networking (ICDCN '23).

- A structured inverse-hierarchy approach to extracting and explaining information from a new dataset of unknown design specifications with respect to the information encoded in the dataset used to train a deep classification model. This contribution is described in the following publication:

– **Alevizaki, A.**, Pham, N., Trigoni, N. (2024). Revealing semantic mappings across HAR datasets. In the 20th Annual International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT 2024).

In summary, this thesis contributes novel smartwatch-based localisation and activity recognition datasets and systems, and provides guidelines on how to connect them with similar yet semantically inconsistent target domains.

1.5 Thesis outline

The remaining of this thesis is outlined as follows: in Chapter 2.7, we provide an overview of related literature. Chapter 3.4 introduces the two datasets we contribute in this work, WatchBLoc and watchHAR, and discusses the challenges associated with data collection. Chapter 4.7 proposes RoomE, an algorithm for room-level semantic localisation using ambient IoT signals and inertial data from a smartwatch. Chapter 5.7 then demonstrates CHiHAR, a hierarchical approach to recognising activities of daily living, enabling a centralised approach to providing similar information to different mobile applications exactly as necessary by each of them. Chapter 6.7 studies the challenges in crowdsourced dataset collection and proposes SeMEDA, an inverse-hierarchy approach to semantic understanding of a crowdsourced dataset and its semantic alignment with another known dataset. Finally, Chapter 7.2.4 concludes this thesis and discusses challenges and future directions.

Chapter 2

Background

2.1 Introduction

In this chapter, we introduce some fundamental concepts relevant to the work in this thesis. We analyse the availability of datasets related to human indoor positioning and activity recognition and emphasise the necessity of high-quality data to drive progress in this research field. We also provide an overview of existing approaches in the fields of human indoor positioning, human activity recognition and knowledge discovery, the concepts that we explore in this thesis in an attempt to establish informative semantic context for human behavioural patterns, encompassing factors like location, activity, and the situational nuances that may allow for controlled or naturalistic behaviours.

2.2 Preliminaries

In this section, we will explore some key concepts that are relevant for a reader to comprehend this thesis. We will highlight relevant sensor modalities for data collection that are pertinent to this research, along with popular signal data processing techniques and learning mechanisms.

2.2.1 Sensors and systems

Our interconnected world offers an extraordinary platform for crafting intelligent systems that leverage a variety of sensor modalities; radio signals, inertial, visual and LIDAR point cloud data, as well as ambient signals such as Wi-Fi or magnetic interferences, can be easily recorded, as humans are often equipped with a combination of such sensors. With this vast amount of available sensor data, we are in a unique opportunity to train systems that can capture elements of human behaviour as a user interacts with their environment.

2.2.1.1 Sensor data processing

While raw sensor data are precious due to their rich embedded information, they often contain noisy, erroneous or missing data. As such, it is important to ensure consistent and clean data before utilising them in their raw form, or, as is often necessary, transform them into other, more explainable feature representations [48, 49].

There are three key preprocessing steps for data preparation: i) denoising, ii) handling missing data, and iii) identifying outliers [50, 51]. This process is the first step in any scenario of learning models from data, both when utilising raw data directly and when engineered features are employed.

As this thesis focuses on ambient and inertial data, this section focuses on data preprocessing for these modalities. However, the same principles apply for other sensor types.

Raw sensor data collected from IoT devices often contain significant amounts of noise and irrelevant information. Noise removal is commonly performed by applying appropriate filters to data, such as moving average or moving median filters [52], or low-pass filters to remove high-frequency noise; continuous and discrete wavelet transforms have also been effectively utilised for signal denoising, by thresholding of the calculated wavelet coefficients [50, 53].

Similarly, low-cost IMUs are susceptible to various error sources arising from mechanical, electronic, and signal processing imperfections. Existing approaches

for mitigating noise in IMU measurements and compensating for intrinsic model inaccuracies predominantly rely on statistical techniques. A common strategy in inertial navigation involves filtering IMU data to suppress or remove high-frequency components [54]. Auto-regressive and moving average (ARMA) methods are widely employed in this domain, similar to the IoT signal denoising mechanisms discussed earlier [55]. Beyond noise, IMU readings are also influenced by sensor biases, typically represented as additive terms in measurement equations [56].

Missing data can be handled in various ways, from simple techniques such as substituting missing values with the mean signal value of existing datapoints [57] or utilising clustering and nearest neighbour to decide on the missing values [50, 58], to linear, cubic, multinomial or more complicated interpolation approaches [58, 59], as well as probabilistic techniques, such as probabilistic matrix factorisation [60].

Finally, outlier detection can be established by training classifiers for standard and outlier values [61] (borrowing from the field of anomaly detection), principal component analysis [62], or voting mechanisms, when more than one nodes are available to correct erroneous values [63].

2.2.1.2 Signal processing for feature extraction

Following data preparation based on the sensor data processing techniques discussed above, it may be necessary to transform raw data to alternative feature representations. Here, we investigate some key signal processing techniques towards this direction that are relevant in this research.

Short-time fourier transform The short-time fourier transform (STFT) is a signal processing technique that investigates the changing frequency content of a signal over time. For a signal $x(t)$, the STFT is calculated by segmenting the signal into short, overlapping time segments $x_w(t) = x(t) \cdot w(t)$ and computing the Fourier transform $X(t, f)$ for each segment:

$$X(t, f) = \int_{-\infty}^{\infty} x_w(\tau) \cdot e^{-j2\pi f\tau} \, d\tau. \tag{2.1}$$

A useful representation of the STFT is the power spectrogram, which represents the squared magnitude of the STFT:

$$|X(t,f)| = \left| \int_{-\infty}^{\infty} x_w(\tau) \cdot e^{-j2\pi f\tau}\, d\tau \right|. \tag{2.2}$$

Wavelet transform The wavelet transform is a time-frequency signal processing method that is very effective at capturing patterns in the signal that are highly correlated in structure with the analysing wavelets ($\Phi_{(s,l)}$, $\varphi_{(s,l)}$), which are defined as,

$$\Phi_{(s,l)}(x) = 2^{\frac{s}{2}}\Phi(2^{-s}x - l) \tag{2.3}$$

$$\varphi_{(s,l)}(x) = 2^{\frac{s}{2}}\varphi(2^{-s}x - l) \tag{2.4}$$

where s, l, Φ, and φ are the scaling and dilating factors and wavelet and scaling generation functions, respectively. At each decomposition level of the transform, the input signal ($f(t)$) is decomposed into approximation (W_a) and detail coefficients (W_d) as,

$$W_a(s,l) = \frac{1}{\sqrt{N}}(\sum_t f(t)\varphi_{(s,l)}(t)) \tag{2.5}$$

$$W_d(s,l) = \frac{1}{\sqrt{N}}(\sum_t f(t)\phi_{(s,l)}(t)) \tag{2.6}$$

These coefficients can be represented as a scalogram, a 2D visual representation of the coefficient values across different scales for each point in time.

There are infinite mother wavelets available, and different mother wavelets will produce different results when applied on the same signal [64]. As such, choosing the appropriate set of wavelets for feature extraction is essential to achieve a robust transformation from the raw data to another representation domain.

2.2.2 Machine learning

Machine Learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computer systems to improve their performance on a specific task through learning from data without being explicitly programmed.

The key components of machine learning encompass supervised learning, where models are trained on labelled datasets, and unsupervised learning, where algorithms discern patterns in unlabeled data independently. Feature engineering stands out as a critical concept involving the selection, transformation, or creation of pertinent features from raw data to enhance model performance. The quality of generated features and number of training data are key to building accurate models; overfitting occurs when a model learns the training data too well, including noise and outliers, leading to poor generalisation on new data. Underfitting, on the other hand, happens when the model is too simple to capture the underlying patterns in the data.

2.2.2.1 Feature engineering

Feature engineering is a key component of machine learning and data analysis that involves designing and selecting informative statistics or representations from raw data to enhance the performance of predictive models.

Traditional feature engineering involves carefully crafting and extracting characteristics from data to capture domain-specific knowledge which may not be readily apparent in their raw form. It requires domain expertise to design and select a representative set of features that will achieve robust model performance [65].

The advances in the area of deep learning have shifted the feature engineering problem from handcrafting features to automatically extracting high-dimensional, complex features in the latent space of a deep model's hidden layers. These learned features represent the data in higher-dimensional spaces and capture intricate, abstract patterns and hierarchies within the input data. While this approach removes the domain-specific expertise required in handcrafted feature engineering, it is more computationally intensive and data-demanding, as deep neural networks typically learn from large datasets. The choice between the handcrafted and deep approach to feature engineering thus depends on the specific problem to be solved.

Regardless of the chosen mechanism, the generated features can be used to train machine learning models.

2.2.2.2 Machine learning models

In this section, we will highlight some popular machine learning models, that are used in this thesis. While deep models are preferred nowadays, particularly in the field of human activity recognition, traditional machine learning has been effectively utilised, and is still often employed as baseline or for comparison [40, 41, 66, 67].

Linear Regressor A linear regressor is a statistical and machine learning model that seeks to establish a linear relationship between a dependent variable (also known as the target or response variable) and one or more independent variables (predictors or features). This relationship is typically represented by a linear equation of the form:

$$\bar{y} = \bar{a}^{\mathrm{T}}\bar{x} + \bar{b}, \tag{2.7}$$

where $\bar{y}$ is the dependent variable, $\bar{x}$ is the dependent variable and $\bar{a}$, $\bar{b}$ are the model parameters. The goal of linear regression is to estimate the values of the coefficients $\bar{a}$, $\bar{b}$ that best fit the observed data, minimising the sum of squared differences between the predicted values and the actual values of the dependent variable.

Decision Trees A decision tree is a non-linear supervised learning algorithm used for both classification and regression tasks. It constructs a tree-like model where each internal node represents a decision based on a specific feature, and each leaf node represents an outcome or prediction. Decision trees partition the feature space recursively using criteria such as entropy to make data-driven decisions, making them interpretable and suitable for tasks involving complex decision-making.

Random Forests Random Forests are an ensemble learning method that combines multiple decision trees to improve predictive accuracy and reduce overfitting. Each tree is trained on a random subset of the data, and predictions are aggregated to produce a final result. Random Forests are known for their robustness and high performance in various classification and regression tasks.

Support Vector Machines Support Vector Machines (SVM) is a supervised learning algorithm used for classification and regression tasks. SVM operates by identifying the hyperplane that maximizes the margin between classes in the feature space. The instances closest to this hyperplane are called support vectors. For non-linearly separable data, SVM employs a technique called the kernel trick, where it maps the input data into a higher-dimensional space to make it linearly separable. The selection of an appropriate kernel function significantly impacts the performance of the SVM model. The linear kernel is one of the simplest kernels, and it's suitable for linearly separable data. It defines the decision boundary as a straight line, plane, or hyperplane in higher dimensions. The linear kernel computes the dot product between two vectors, making it efficient for large-scale datasets. The Radial Basis Function (RBF) kernel is commonly used for non-linear data and maps data points into a higher-dimensional space using a non-linear transformation. It is effective for capturing complex decision boundaries. The RBF kernel uses a similarity measure to place more weight on closer points, creating non-linear separation boundaries.

Gradient Boosting Classifier Gradient Boosting is an ensemble learning technique that builds an additive model in a forward stage-wise manner. It combines weak learners (typically decision trees) to create a strong predictive model. Gradient boosting minimizes a loss function by adding decision trees sequentially, with each tree correcting the errors made by the previous ones. It is widely used for both classification and regression tasks.

Gaussian Naive Bayes Gaussian Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem with the "naive" assumption of independence between features. It's particularly effective for classification tasks when dealing with continuous or real-valued features. Gaussian Naive Bayes is a variant of the Naive Bayes algorithm, specifically designed for features that follow a Gaussian (normal) distribution. It's employed to predict the class labels of instances based on the conditional probability of features given the class.

The "naive" assumption in Gaussian Naive Bayes implies that the presence of a particular feature in a class is unrelated to the presence of any other feature, given the class label. This assumption simplifies the probability calculation, and it's particularly effective when this independence assumption approximately holds true for the given dataset.

2.2.3 Deep learning

Deep learning refers to a subfield of machine learning that involves the use of artificial neural networks, particularly deep neural networks, to model and solve complex problems. These networks are characterized by multiple layers (deep architectures) that allow them to automatically learn hierarchical representations of data, extracting relevant features at different levels of abstraction. Deep learning has demonstrated remarkable success in various domains, including image and speech recognition, natural language processing, etc.

Deep neural networks require a substantial amount of training data to effectively learn and discern patterns. The depth and complexity of these networks, with numerous interconnected layers, make them capable of capturing intricate features and representations from data. Adequate training data enables the model to adjust its parameters through iterative learning, improving its ability to make accurate predictions on new, unseen examples. The lack of enough quality data for training can significantly affect a deep model's learning capacity, leading to overfitting. As the availability of large and diverse data is sometimes limited, particularly in supervised domains where annotations are required, mitigating such problems is key for the model to achieve better performance across a range of inputs. These techniques are known as model regularisation; effective regularisation strikes a balance between fitting the training data well and generalising to new, unseen data. It plays a crucial role in preventing models from becoming too complex or too simplistic, thus enhancing their ability to make accurate predictions on diverse datasets. Regularisation techniques contribute to the robustness and reliability of deep learning models in real-world applications.

2.2.3.1 Regularisation approaches

Here, we present a summary of regularisation mechanisms for deep neural networks.

Data augmentation Generating additional training samples by applying random transformations to existing data helps expose the model to a broader range of scenarios.

Dropout This technique involves randomly setting to zero a fraction of neurons during training, preventing the network from relying too heavily on specific neurons and enhancing generalisation.

Weight regularisation Adding a penalty term based on the magnitude (L2 regularisation) or absolute value (L1 regularisation) of the weights to the loss function helps prevent overly complex models.

Early stopping Monitoring the model's performance on a validation set during training and stopping when performance on the validation set starts to degrade helps prevent overfitting.

Batch normalisation Normalising the input of each layer in a network helps stabilise and accelerate training, acting as a form of regularisation.

2.2.3.2 Key deep learning models

Convolutional neural networks (CNN) A convolutional neural network (CNN) is a class of deep neural networks designed originally for data requiring translational invariance, such as images and videos [68]. CNN models perform convolution and pooling operations over the input data, aiming to capture local patterns and learned hierarchies. They have proved excellent feature extractors for spatially correlated data and are thus often used in computer vision.

Reccurrent neural networks (RNN) Recurrent Neural Networks (RNNs) are a type of artificial neural network designed to handle sequential data and capture dependencies over time. Unlike traditional feedforward neural networks, RNNs have connections that form directed cycles, allowing them to maintain a hidden state that retains information about past inputs. This architecture makes RNNs well-suited for tasks involving sequences, such as time series analysis, natural language processing, and speech recognition.

However, traditional RNNs have limitations, particularly in handling long-term dependencies, known as the "vanishing gradient" problem. As the network processes information over many time steps, the gradients used for learning diminish, making it challenging to capture dependencies that are distant in time.

Long short-term memory (LSTM) A long short-term memory (LSTM) network is a class of deep neural networks from the residual neural network family (RNNs) that aim to capture long-range dependencies within sequential data. LSTMs have demonstrated remarkable performance in capturing temporal patterns in sequential data, such as natural language processing [69] and speech recognition [70], owing to their ability to capture intricate temporal patterns and exhibit resistance to the vanishing gradient issue associated with traditional RNNs.

ConvLSTM architecture Combining a CNN with a LSTM network can be a powerful approach in applications involving sequential data exhibiting correlations in the spatial domain [71]. It can be particularly beneficial for time series data: the CNN module can act as a feature extractor, followed by the LSTM cell to capture time dependencies and patterns. ConvLSTM architectures have also been extensively explored in the area of Human Activity Recognition (HAR), including for data from inertial measurement units (IMUs) [72–75]. Most commonly, the data are segmented in windows of 2-to-5-sec length to be prepared as inputs to the models [73, 74]. As CNNs perform particularly well for image-like data, it is a common approach to first handcraft a visual representation from the raw data, e.g., an FFT spectrogram [74] or other composite images [75] to use as input to the models.

In this section, we have presented the key theoretical concepts that are necessary for a reader to comprehend this thesis. Having focused here on methodologies, in the next section, we will discuss the availability of high quality datasets that can be used to train robust models for the problems of human indoor positioning and activity recognition that we study in this thesis.

2.3 Existing datasets

Machine learning algorithms largely depend on good-quality data to be trained and evaluated effectively. While some application domains, e.g., computer vision, have a large number of standardised image datasets for benchmarking (e.g., MNIST, CIFAR-10, MS-COCO, etc.), there are not many publicly available datasets that are suited for the problems addressed in this thesis: human indoor positioning and human activity recognition from smartwatch data.

Regarding indoor positioning methods for the home scenario, in [76] the researchers collect geographical ground truth locations and BLE received signal strength indicators (RSSIs), but all data are collected in a single open space, with known device configurations and floorplan. Similarly, the UJIIndoorLoc-Mag database [77] provides semantic location data and smartphone IMU data but is only constrained to snapshots of corridors and not continuous movement across different rooms. The Miskolc IIS Hybrid IPS dataset [78] offers geographical ground truth location, BLE and Wi-Fi RSSIs, as well as magnetometer data from a smartphone, which constraints the viable positioning methods to those based on magnetic field (which usually require fingerprinting). The dataset collected with the Smart Home in a Box (SHIB) technology [43] is the most complete database, offering semantic location estimates and RSSIs from gateways and IMU data in a wrist wearable in a real home environment, albeit is constrained only to a subset of the house's semantic locations.

Similarly, HAR datasets are primarily focused on data collected in smartphones and often suffer from a limited range of recorded activities. The UCI HAR dataset [79] is a very popular dataset of 6 locomotive activities but only contains data

from smartphone sensors. The Opportunity dataset [44] on the other hand is rich in sensor data, including both smartphone and smartwatch inertial recordings, but the set of recorded activities is focused on low-level actions, such as opening/closing doors or drawers, etc. An important contribution, as it includes data from both smartphones and smartwatches recorded in-the-wild, is the ExtraSensory dataset [47]; however this naturalistic approach to data collection can lead to missing labels and inconsistencies to the annotations, as well as unbalanced data classes, as the authors themselves report [80]. The PAMAP2 [45] and WISDM [46] datasets are the only available, to our knowledge, datasets that provide accurate smartwatch recordings and capture a rich set of interesting activities of daily living.

2.4 Human positioning

Human Positioning is the area of research that aims to estimate a human's position in a given environment. The position estimate can vary from accurate spatial coordinates (*geographic localisation*) to semantic location (*semantic localisation*) or even orientation of the pose of the individual (*pose estimation*), depending on the task in question. The high availability of Global Positioning System (GPS) has achieved high accuracy and reliable tracking in real-time. Indoor positioning, on the other hand, is more challenging due to a number of factors, including but not limited to signal attenuation and fading, cost of infrastructure, and privacy. Here, we will examine key approaches to human indoor positioning using a variety of infrastructures and sensor modalities.

Note that the primary goal of this thesis is to discern continuous human behavioural patterns while respecting the user's privacy and natural conduct. Consequently, approaches that necessitate intricate and costly infrastructure, disrupt the user's unrestricted movements, or pose privacy concerns are not appropriate for our research focus. Therefore, this chapter centres on exploring lightweight, scalable solutions for indoor human positioning and will thus focus on radio-frequency-based and inertial navigation approaches. Nevertheless, we will provide a complete overview of existing positioning solutions renowned for their accuracy in different

contexts, explaining why they are unsuitable for indoor human positioning in a naturalistic setting.

2.4.1 Positioning methods

In this section, we will initially present the fundamental tools underpinning positioning methodologies. The concepts discussed herein are not inherently exclusive to a specific sensor type or positioning system; rather, they can be applied across various infrastructure scenarios. Nevertheless, certain techniques may exhibit greater suitability for specific technologies, as we will examine in Section 2.4.2.

2.4.1.1 Signal measurements

Let us first define some key concepts that are relevant for estimating the distance between a transmitter and a receiver in a positioning system.

Time of Arrival (ToA)* or *Time of Flight (ToF) This method performs ranging based on the relationship between transmission time, speed and distance. It involves calculating the variance between the received signal's time (as per local clock measurements) and the transmission time (based on global time measurements), or vice versa, as described in [81]. This nuanced approach, encompassing both local and global measurements, necessitates precise time synchronisation among network nodes, which can introduce complexity into network design. Knowing the ToA and the propagation speed of the signal, we can estimate the equivalent distances between transmitter and receiver drawing upon fundamental principles of physics, i.e.:

$$d = cT, \tag{2.8}$$

where d is the calculated range measurement, c is the propagation speed of the signal, and T is the ToA, as computed by the sensors.

Time Difference of Arrival (TDoA) Much like the Time of Arrival (ToA) method, Time Difference of Arrival (TDoA) relies on transmission times for distance estimation. However, in situations where the exact emission time remains unknown, TDoA offers a solution by enabling the computation of distances. This is achieved by translating the disparity between the recorded time of signal arrival from a pair of reference nodes, as outlined in [82], into the difference in range estimates for those reference nodes. To determing the transmitter's position, we need to solve a set of equations similar to Equation 2.8:

$$\Delta d_i = c\Delta T_i, \tag{2.9}$$

where Δd_i is the calculated range measurement between the i^{th} receiver and a reference receiver, c is the propagation speed of the signal and ΔT_i is the TDoA between the i^{th} receiver and a reference receiver, as computed by the sensors.

Angle of Arrival (AoA) The AoA is defined as the angle formed between the propagation direction of an incoming wave and a designated reference direction, known as orientation. This orientation is typically represented in degrees and follows a clockwise direction from the North. When the orientation aligns with 0^o, indicating a Northward direction, the AoA is considered absolute; otherwise, it is relative in nature [83]. The fundamental concept behind AoA revolves around the notion that the signal's delay as it reaches two antennas varies based on the transmitter's direction. By measuring the phase difference between the received signals at these two antennas, we can precisely determine the direction from which the signal originates. In essence, if we possess the knowledge of the angles at which a signal is received by two receivers with established positions, we can compute the position of the unknown node (i.e., the emitter) as the third point of a triangle. This triangle has one known side (the distance between the two receivers) and two known angles (the AoAs).

Received Signal Strength (RSS) In indoor environments, the quality of electromagnetic signals as it travels between the transmitter and the receiver can be affected due to several factors; for example, whether there exists a direct line-of-sight (LOS) between the nodes can lead to noticeable variations of the signal at the receiver. Non-line-of-sight (NLOS) can further exacerbate other challenges, such as multipath fading and signal attenuation: as the signal travels through windows, walls, and other significant obstacles, it can be diffracted, reflected and faded, adding variability to the signal strength.

RSS-based positioning aims to alleviate this problem by assuming an attenuation model for signal propagation, specifically a path-loss model, and different assumed models can lead to varying results. The most common wireless signal propagation model is the log-distance path loss model:

$$r = r_0 - 10n \log_{10}(d), \tag{2.10}$$

where n is the path loss exponent, d is the distance between the transmitter and the receiver, and r_0 is the path loss at a reference distance of 1 metre. By applying this model, it becomes possible to estimate the distance between the transmitter and the receiver based on the observed RSS value.

2.4.1.2 Positioning algorithms

Trilateration This method serves as a crucial second step in all the previously mentioned approaches. While Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA), and Received Signal Strength (RSS) methods offer distance estimations between the transmitter and receiver, they alone cannot pinpoint the exact location. Instead, they provide a geometric locus or a subspace within the coordinate system where the node might be situated. To address this challenge, trilateration takes the distances calculated through ToA, TDoA, AoA, or RSS and transforms them into a geometric problem. In a 2D plane, typically used for localising humans, a minimum of three range measurements is necessary. In the case of ToA, each distance from the transmitter represents a circle with a radius equal to

the range measurement around the respective receiver. The node then determines its position by identifying the intersection point of these circles. Similarly, in the case of TDoA, each time-difference estimate yields a hyperbola representing the potential positions of the node. The transmitter's unique location can be determined by intersecting two such hyperbolas. The AoA and RSS methods are commonly also converted into one of these geometrical frameworks based on whether Time of Arrival (ToA) or Time Difference of Arrival (TDoA) can be inferred.

Fingerprinting Fingerprinting stands as one of the most widely employed indoor positioning methods [84]; it comprises two distinct phases: the offline phase and the online phase. The offline phase is a training phase, during which a database of empirical pairs $(coordinates, RSSI)$ is meticulously created. This collection of database entries is called the fingerprinting radio map. During the online phase, signal intensities from unknown positions are recorded and compared with the entries in the fingerprinting radio map using a proximity measure, e.g., k-nearest neighbours (k-NN). The result of the proximity algorithm, combined with the appropriate signal propagation model for each particular task, can yield a re-estimate of the user's location.

Proximity-based positioning Proximity location methods involve the utilisation of emitter nodes to detect instances where the RSS at the receiver exceeds a predefined threshold, indicating that the receiver is in proximity to the emitter. In cases where multiple sensors register readings above the threshold, the node with the highest RSS is designated as the closest node. By utilising multiple threshold boundaries, assets can be categorised within immediate, near, and far proximities from the emitter.

Pedestrian Dead Reckoning (PDR) Pedestrian Dead Reckoning (PDR) is a localisation technique that estimates the position of an individual by integrating data from inertial sensors, such as accelerometers and gyroscopes, with information about the user's starting location [85]. As the pedestrian moves, the system

continuously updates the position based on the user's step counts, stride length, and direction changes. While this method can provide accurate positioning, it is prone to cumulative errors over longer periods of time [86]. Hence, it is necessary to establish mitigation mechanisms that allow for error-correction.

2.4.2 Positioning systems

In the previous section, we outlined fundamental concepts that underpin indoor positioning, independent of specific infrastructure or sensor types. In this section, we will explore current research on indoor positioning, focusing on the primary sensing technology employed.

2.4.2.1 Vision-based positioning

Vision-based approaches to positioning estimate user location by analysing images taken by cameras. While these are most commonly computer vision approaches are employed, as images are taken from standard (RGB) or more recently depth (RGBD) cameras, the increased popularity of augmented reality experiences has drawn interest to estimating user position within the hybrid virtual and real space.

Computer vision The area of computer vision has traditionally focused on developing algorithms and techniques to identify and track objects in images or in videos over time [87]. With the advances in the area of autonomous driving, pedestrian tracking has seen remarkable progress [88], paving the way to identifying and tracking humans from image-like data. In terms of indoor positioning, we can identify two approaches: a passive approach, where the user wears a camera that captures instances of the surrounding environment from the user's viewpoint; and an active approach, where cameras are pre-installed in the environment the user needs to be monitored [89].

In the active approach to vision-based positioning, a number of directions have been explored. Traditional computer vision techniques have been utilise to detect and localise individuals carrying a marker, for example, wearing a certain colour or carrying a particular object [90, 91]. Towards a lighter approach for the user

being tracked, other research directions have explored recursively detecting the foreground or background of the retrieved images and separating the user from that space [92, 93]. More recently, deep learning approaches have been explored to achieve positioning from fixed cameras by estimating the location of body parts [94], full pose estimation [95, 96] or landmark identification [97], for the case of indoor spaces where space structure can be inferred. Note that for these research directions, a map of the environment and the location of the cameras within the environment are required.

The passive approach to vision-based positioning is related to visual odometry (VO): VO estimates the ego-motion of a camera by identifying known locations in its environment or exploiting geometric constraints from consecutive images [98] by extracting a variety of features such as saliency maps, photometric features, etc. [99, 100]. Employing an object-detection approach, standard computer vision techniques have explored identifying marker-style objects in predefined locations in the environment, most commonly QR codes [101, 102]. In a similar direction, many research works have explored creating databases of key landmarks existing in the environment and then estimating the user's position with respect to known static objects shown in the images taken by the worn camera [103–105]. A pioneering approach in the passive-tracking scenario is the use of Simultaneous Localisation and Mapping (SLAM, commonly called Visual SLAM in the context of camera-based positioning), which aims to tackle the accumulated error drift over time by constructing a world map in parallel with pose estimation. Additionally, a loop closure detection module is integrated to identify previously visited locations, facilitating the joint optimisation of pose estimation and map construction. There are many robust Visual SLAM approaches, the most notable being ORB-SLAM [25], a feature-based system operating in real-time with an impressive location estimation error of less than 4cm, and LSD-SLAM [106], that tracks camera pose and builds environment maps using photometric features. StructSLAM [107] interestingly took into account the structural regularity of man-made building environments and detected structure lines along dominant directions. Towards landmark identification, Finally, DynamicSLAM [108] was proposed to perform localisation and mapping

within dynamic environments by distinguishing between static and dynamic objects that appear in the retrieved images and discarding the information from dynamic objects that may have moved from their previous known location.

However, the utilisation of cameras imposes disruptions to the user's natural behaviour, particularly for passive approaches, that demand the user to constantly carry the camera. At the same time, there are concerns related to the privacy of both individuals captured within the camera's recordings and the user being tracked in active tracking scenarios. Consequently, employing such methods proves to be impractical for impractical for human indoor localisation, particularly in privacy-preserving environments, e.g., homes or private office spaces.

Augmented reality (AR) With the advances in augmented reality, human positioning within a hybrid real and virtual space has also become an area of research interest. More commonly, AR approaches are based on landmark identification within a pretrained dataset: AR systems overlay virtual objects with the real world and then use these markers for the purpose of positioning, tracking and navigation [109, 110]. SLAM-based approaches are also prominent, as it is necessary for the virtual environment to be reconstructed and accurately overlayed on the real-world [111, 112].

2.4.2.2 Sound-based positioning

Acoustic signals have been used for positioning early in research, as they propagate through air and other materials with known properties at a much slower speed than other signals, e.g., electromagnetic. Hence, it is possible to measure the time needed for a sound signal to travel between a transmitter and a receiver, allowing for various ranging-based methods, such as ToA, TDoA, etc. While both audible and ultrasonic sounds have been explored, ultrasonic approaches are preferable as they are not detectable by the human ear [89].

Ultrasonic positioning Ultrasonic location-based systems use sound frequencies higher than the audible range (beyond 20 KHz) to determine the user position using the time taken for an ultrasonic signal to travel from a transmitter to a receiver. Their speed of travel, negligible penetration in walls and low cost of transducers make it interesting for studies in indoor positioning. The most prominent ultrasonic positioning systems are Active Bat [113] and Cricket [114]. Active Bat consists of an array of fixed microphones installed on the environment's ceiling and a tag carried by the user. The tag broadcasts ultrasonic signals while the receiver processes the signals one at a time to prevent interference and estimates the tag's (hence also the user's) position using ToA and trilateration. Hence, it requires at least three microphones for effective localisation. Cricket employed the opposite approach: a set of ultrasonic beacon emitters were installed in fixed locations in the environment, with the user carrying the receiver instead. Cricket is a hybrid system, also utilising radio-frequency (RF) signals, and estimates user location based on their proximity to the nearest beacon. More recently, the work in [115] used an array of slow-moving ultrasonic sensors using Doppler shift compensation to improve location estimation, reporting location accuracy at less than 10 mm. However, this work is limited by low movement speeds for the array over short distances that maintain line-of-sight conditions.

Audible sound Unlike ultrasound positioning systems, which have a limited range because of high attenuation while propagating in the air, audible sound approaches are not limited in range. Most commonly, audible sound systems utilise a mobile device that emits a sound to be sensed by microphones installed in the environment to estimate the user's position with ToA. The most common audible systems following this principle are Beep [116], BeepBeep [117] and Guogo [118].

While accurate positioning can be achieved with sound-based systems, most approaches require cumbersome and expensive infrastructure that is challenging to deploy and maintain. At the same time, sound propagation suffers from multipath effects such as noise, reflection and interference, causing signal attenuation. Finally,

the use of microphones can raise privacy concerns, both in terms of the content of recordings at the receiving microphones, as well as for identity protection, as voice is an identifiable human feature.

2.4.2.3 Light-based positioning

Infrared (IR) technology Towards a more privacy preserving approach, infrared technology (IR) in Indoor Positioning Systems (IPS) utilises electromagnetic radiation with wavelengths beyond the visible light spectrum [89]. IR systems, such as the ActiveBadge and VICON systems, have proved highly accurate in indoor positioning, achieving estimating a user's position with a less than 10cm error [119, 120].

Nonetheless, they usually require the user to carry markers to achieve accurate location estimation, hence impeding comfort and natural conduct. At the same time, the installation of IR systems is expensive, making it challenging for the systems to generalise as a standardised solution to indoor positioning [121].

Light Detection and Ranging (LiDAR) LiDAR systems use laser light waves to measure distances and generate point clouds (i.e., a set of 3D points). The distance is computed by measuring the time of flight of a light pulse, while the direction of a transmitted laser is tracked by gyros. By matching the measured point cloud with that stored in a database, an object can be located. LiDAR can be employed to create detailed maps of indoor spaces, and the data collected can be used for accurate localisation and tracking of objects or devices. It is particularly useful in environments where other positioning technologies may face challenges, such as limited line-of-sight or signal interference. In this direction, the work in [122] explored robot positioning in indoor spaces using LiDAR as the sole sensing modality for SLAM, with accurate positioning estimates. Similarly, the authors in [123] explored a map-matching approach from architectural floorplans to SLAM maps of the environment using LiDAR measurements. However, LiDAR systems are particularly expensive (around $1000 for a short-range sensor) [26] to be used in private indoor spaces.

Visible Light Communication (VLC) VLC is an technology for high-speed data transfer that uses visible light between 400 and 800THz, modulated and emitted primarily by Light Emitting Diodes (LEDs) [124]. Such approaches are also primarily based on ToA, measuring the distance from the LED transmitter to a device capable of capturing light intensity (e.g., photocell) or an image sensor (e.g., camera) [125–127]. The advantage of an image sensor is that it can register simultaneously several lamps with their positions, thus achieving a more precise location estimate.

2.4.2.4 Radio-frequency-based positioning

Radio-Frequency-based (RF-based) localisation is a positioning technique that relies on the use of radio waves to determine the location or coordinates of an object or device in a given area or space. This technology works by measuring the time it takes for radio frequency signals to travel between a transmitter (or multiple transmitters) and a receiver.

Ultra-Wideband (UWB) Ultra-Wideband positioning (UWB) operates by transmitting nano-pulses ($< 1\ nsec$) to a very large bandwidth ($> 500MHz$ wide). UWB is capable of providing very accurate positing estimates (up to a $1cm$ level accuracy) in real-time due to its high multi-path resolution and high-speed data transmission owing to its large bandwidth [128, 129].

In a passive approach to UWB positioning, the position of a person or object is determined through signal reflection, not via an attached UWB tag. When individuals move within a space with known UWB transmitters and receivers, their bodies reflect signals emitted by the transmitters. Receivers detect these reflected signals, and TOA, TDOA, and trilateration techniques estimate the person's position. On the other hand, active UWB-based positioning systems use battery-powered UWB tags carried by the user that transmit ultra-short UWB pulses to fixed UWB sensors [130–132]. Commercial system Ubisense Real-Time Location System [133] based on this approach claims an accuracy of 15 centimetres. However, it requires dedicated infrastructure and raises privacy concerns, as the user's location is disclosed to the system.

The application of UWB in indoor environments offers advantages such as long battery life, robust flexibility, high data rates, penetrating power, low power consumption, and high positioning accuracy with minimal interference and multipath effects [134]. Strategic placement of more UWB sensors could widen coverage, enable real-time tracking, improve positioning accuracy, and reduce the impact of signal impairments [130, 131]. However, scaling UWB can be expensive due to the need for deploying more sensors across a wide coverage area to enhance performance [131]. The deployment of anchor reference points is also labour-intensive and requires careful calibration, making it challenging to install in private indoor spaces [89].

Wi-Fi Wi-Fi is a wireless communication protocol operating in the 2.4-GHz Industrial, Scientific and Medical (ISM) band. Nowadays, the ubiquity of Wi-Fi access points in today's world presents a remarkable opportunity; it allows us to harness the RSSI values from Wi-Fi signals at various points within a given space to calculate the receiver's precise position relative to these Wi-Fi access points. This opportunity has seen tremendous growth, especially in light of the widespread adoption of smartphones and other intelligent devices equipped with the capability to sense ambient sensor signals [89, 124, 135].

The majority of Wi-Fi positioning methods rely on trilateration or fingerprinting techniques [135, 136]. However, fingerprinting relies heavily on a consistent signal propagation model, which is challenging to establish accurately in indoor spaces. The computational complexity of proximity algorithms in large indoor spaces can also be a concern. To address these, [137] creates a local propagation model for each subarea in the environment, reducing computational demands. Another notable challenge is device dependence, where using different sensing devices for offline and online phases can lead to low-confidence estimates. To overcome device-dependent fingerprint radio maps, [29] standardises fingerprint maps across devices, employing statistical shape analysis and weighted k-NN (WkNN) for improved accuracy. In the same direction, [138] introduces neighbor-relative RSS (NR-RSS) values instead of

absolute RSS (ARSS). This method encodes movement information using a Markov Prediction Model (MPM), ensuring energy-efficient tracking of the device bearer.

Even though there has been extensive research on Wi-Fi positioning, Wi-Fi consumes significant power since its original goal was not localisation but to achieve communication through signal propagation in long distances [139]. In indoor spaces, however, this is not really required; with the development of the low-power energy-efficient BLE, work has been done towards examining the potential of BLE beacons as Wi-Fi substitutes [139, 140].

Bluetooth Low Energy (BLE) BLE is a wireless communication technology designed for short-range communication with low power consumption. BLE is a variation of the classic Bluetooth technology, optimized for applications where energy efficiency is crucial. It has gained significant popularity in indoor positioning systems due to its energy-efficient characteristics and suitability for indoor environments.

BLE better relates RSSI to distance and improves localisation, compared to Wi-Fi [139, 140]. Several Wi-Fi and BLE indoor positioning approaches have since emerged; fingerprinting methods have traditionally been a classic approach [29, 137, 138, 141, 142]. Though most methods try to tackle various problems arising from the fingerprinting technique, e.g., the size of the fingerprint map or the map's dependency on the device, most fingerprinting methods require a map of the environment [143] and a well-defined path-loss model [144], which can be tricky to acquire or establish, particularly in private indoor spaces, e.g., homes.

Radio-Frequency Identification (RFID) RFID is a wireless communication technology for automatic identification involving electromagnetic wave transmission between RFID readers and RFID tags to track objects [134, 145]. The RFID reader queries and reads data from tags, which respond with unique identification or stored information. RFID tags are classified as active or passive [134, 146]. Active tags, powered by batteries, have a broader transmission range, requiring fewer tags. Passive tags, without batteries, have a shorter range, drawing power from the reader's signal [146].

Active systems involve stationary active tags and a mobile reader, offering wider coverage with fewer tags, but at a higher cost due to expensive active tags [147–149]. Passive systems use stationary passive tags and a mobile reader [145, 150, 151]. Notably, the authors in [145] present a unique approach with a stationary reader and mobile passive tags, eliminating the need for tag batteries and reducing system costs.

2.4.2.5 Sensor-based positioning

Inertial Navigation Systems (INS) Inertial navigation involves estimating a future position based on an initial position, speed, and direction. Traditionally, this technology relied on expensive and highly precise inertial measurement units (IMUs). However, recent advancements in micro-electromechanical (MEMS) technology have transformed IMUs by making them more affordable, compact, and energy-efficient. Nowadays, IMU sensors are readily available in common smart devices such as smartphones, tablets, and smartwatches. These devices are typically carried by individuals throughout the day, making them valuable sources of mobility data [138, 152], encoded in their inertial sensors (accelerometer, gyroscope, magnetometer). By integrating these measurements over time, we can identify a pedestrian's step count, step length, and direction of movement. This is a method known as Pedestrian Dead Reckoning (PDR): starting from an initial known position, PDR continuously updates and estimates the pedestrian's new position as they move, based on their steps and orientation changes. However, IMU data from smart devices can be inherently noisy due to the various ways these devices are held and used, leading to multiple error sources, including sensor noise, walking noise, thermo-mechanical noise, scale factor variations and axis misalignment [153]. Estimating the direction of movement in such noisy data can be challenging; still, their low-cost and ubiquity have led to significant indoor positioning research utilising inertial sensors. There are two key approaches to PDR: strapdown navigation, and step and heading estimation.

In the strapdown technique, an IMU sensor is affixed with its three principal axes aligned to the orthogonal axes of the tracked rigid body, ensuring no relative motion for accurate measurements [154]. Most approaches utilise the global reference

frame which involves subtracting gravity and double integrating signal results for three-dimensional space position changes [155, 156]. Integration of accelerations yields distance estimates, and integration of angular speed infers heading [157]. However, double integrations may lead to quadratic error propagation, causing inaccuracies or offsets to result in drift, a deviation from ground truth [157]. Zero-Velocity Updates (ZUPT) can mitigate drift, especially in Inertial Location and Navigation Systems (ILBS) for human assets, achieved through foot-mounted IMUs in sync with the human gait cycle. Human gait, characterised by a stance and swing phase, serves as a regular stationary point, with approximately 60% in the stance and 40% in the swing phase [158]. Yun et al. proposed using gait analysis, specifically the stance phase, for drift correction by resetting velocities to zero when detecting this phase [155]. ZUPT detects stationary points, calculates drift, and corrects IMU velocities, improving accuracy. In literature, ZUPT, combined with an extended Kalman filter (EKF), has demonstrated high accuracy in foot-mounted IMU inertial location systems [159].

The step and heading technique, reliant on IMU accelerometer features, is designed for detecting human steps. Combining the estimated step length with heading information facilitates location estimation only with linear errors, reducing location error propagation to two dimensions. However, variations in stride length introduce additional errors[157]. To address this, RF technologies are commonly employed. Some studies leverage smartphones for location services, combining phone Wi-Fi and inertial signals for pedestrian tracking alongside step and heading data [160]. The authors in [161] implemented a step-detection solution with BLE beacons, considering various phone positions. The system determines triangulation feasibility based on beacon count visibility; otherwise, a fingerprinting approach is used, yielding a reported accuracy of 1–1.5 m. Additionally, the authors in [162] proposed an IMU service with time of arrival calculations from ultrasound pulses, incorporating a maximum a posteriori (MAP) algorithm for determining the most likely next location based on the previous state.

Magnetic field Magnetic positioning systems utilise magnetic signals for determining positions within a magnetic field [163–165]. These systems typically comprise fixed transmitters and receivers mounted on users or tracked objects. Receivers capture magnetic signals and relay position information to a centralised location to estimate the transmitter's location. Other approaches move away from dedicated devices, leveraging the magnetic properties of existing structures like pillars, steel elements, and electronic appliances in the monitored environment instead [166].

Magnetic positioning involves two main magnetic field types: static magnetic fields and low-frequency alternating magnetic fields. Static or artificial magnetic fields are generated by permanent magnets or coils using Alternating Current (AC) or pulsed Direct Current (DC). Low-frequency alternating or electromagnetic fields are formed by static charges producing electric fields and currents generating magnetic fields, leading to the creation of electromagnetic fields [166]. A number of research works have focused on magnetic fields from currents [163], enhancing tracking speed through the excitation of magnetic sources and analysing resultant sensor output.

Towards optimising sensor placements, a number of research works aim to improve accuracy and coverage area while minimising costs. Systems using permanent magnets have been explored, focusing on magnetic localisation for tracking wireless capsules [167], while the authors in [164] proposed a 3D indoor positioning system based on artificial magnetic fields, using active magnetic coils and passive magnetometers.

Fingerprinting approaches are also prevalent in magnetic positioning. The authors in [165, 166, 168, 169], leverage magnetic field variations for position estimation. These systems often involve server-based comparisons of magnetic fingerprints for accurate positioning and navigation. However, challenges include errors, increased costs, and extended configuration times, particularly for larger coverage areas [165, 170].

2.4.2.6 Hybrid approaches

The conventional methods for indoor localisation point through a geographical viewpoint. However, it becomes apparent that in many cases, especially within private indoor spaces, the precise tracking of a user's path or step count is of limited interest to most applications. Instead, the focus has shifted towards determining the user's semantic position within the environment, as this information proves to be sufficiently informative and also contributes to enhancing privacy protection: it relaxes the need for sensitive user details like height or leg length, which are typically essential for accurate Pedestrian Dead Reckoning (PDR) systems to estimate step detection and step length. Similarly, it alleviates the requirement for intricate floorplans of the house, which are indispensable for traditional map-matching approaches.

A noteworthy algorithm in the area of semantic localisation of users introduced an algorithm based on human-robot coexistence to estimate the user's room-level location based on smartphone human activity recognition (HAR) and human-robot colocations [171]. This innovative approach seeks to estimate the user's location at the room-level granularity by leveraging smartphone-based Human Activity Recognition (HAR) and monitoring instances of human-robot colocations. An intriguing aspect of this methodology is its unique approach to establishing ground truth. Instead of relying on external ground truth data, the authors strategically place BLE beacons within the target rooms. The user's location is then inferred based on which beacon emits the highest Received Signal Strength Indicator (RSSI) value at any given time.

An alternative approach that focuses solely on human-centric perspectives is S-Smart [172]; this combines recognition of human activities with PDR to dynamically learn semantic positions of interest around the house, e.g., windows, doors, etc. However, It requires a relatively dense on-body sensor set-up, including a smartphone in the pocket, a wrist-worn IMU, and a foot-mounted sensor, which are inconvenient for everyday use. On a similar note of using HAR to infer semantic locations around the house, the AtLAS system [173] fuses HAR and PDR from smartphone

IMU data and Wi-Fi RSSIs to both locate landmarks in the environment and to localise the user in it. However, it's important to highlight that the AtLAS system relies on Wi-Fi fingerprinting, a technique that demands meticulous calibration and infrastructure deployment. In a similar direction, the researchers in [174] establish a robust correlation between room-level locations and corresponding RSSI values, enabling accurate and context-aware user localisation. What sets this approach apart is its adoption of a supervised learning paradigm. Through this approach, the researchers establish a robust correlation between room-level locations and corresponding RSSI values, enabling accurate and context-aware user localisation.

A fusion of ambient RF signals and inertial data has been widely explored and successfully used in indoor public spaces (e.g., shopping malls) that are equipped with large wireless sensor networks (WSN/WLAN) with numerous access points (APs). These methods usually employ Wi-Fi APs or Bluetooth low energy (BLE) beacons [29] together with smartphone inertial measurement unit (IMU) sensors [30] to achieve an accuracy with an estimated $2 - 10$ metres error [30]. However, these cannot easily generalise for the scenario of positioning in private indoor spaces, where errors beyond 4 m would likely correspond to a completely different room in a house. At the same time, they usually require extensive fingerprinting or crowdsourcing [175], which is not an option at home, and are developed under the scope of smartphone sensors; while smartphones are a competent choice for positioning in public spaces, this is not necessarily the case for private indoor spaces, as people do not usually carry the device with them at all time at home or in their working space; rather, the device typically lies in a surface nearby, hence inhibiting accurate and continuous tracking of the user's location.

2.4.3 Summary and challenges

In this section, we have explored and discussed various methods found in the literature related to the problem of indoor positioning. Our examination has encompassed a wide spectrum of techniques, ranging from visual positioning to radio-based and inertial positioning methods. We have analysed the types of sensors

used and the constraints different environments pose, highlighting the need for different approaches in each environment. While many of these solutions offer high accuracy, it has become evident that a significant number of them are characterised by complexity and costliness, especially when considering the problem of positioning individuals in private indoor spaces.

With this realisation, our focus has shifted towards the utilisation of wearable sensors, such as smartphones, smartwatches, etc., which offer the advantages of being lightweight and readily available. Throughout our analysis, we have placed particular importance on the development of methods that respect user privacy and do not disrupt the user's surroundings. In this context, we have emphasised the significance of ambient sensor signals, which eliminate the necessity for additional infrastructure.

We have underscored the importance of avoiding techniques reliant on fingerprinting or extensive calibration and environment knowledge. While there is a wealth of research concerning indoor positioning in public spaces, the exploration of solutions for private indoor environments remains relatively limited. In this context, approaches involving WiFi and Bluetooth Low Energy (BLE) in conjunction with smartphones have gained prominence for achieving indoor positioning, as they require minimal additional infrastructure. However, the use of smartphones in such scenarios presents its own set of challenges, particularly in terms of continuous monitoring of users within private indoor spaces. As a result, we have concluded that smartwatches represent a viable and promising alternative for addressing these challenges, despite being relatively understudied in the current body of literature. Their potential to enhance indoor positioning methods in both public and private indoor spaces deserves further exploration and investigation.

At the same time, most methods utilise environment-specific information, such as semantic maps of the environment, to provide accurate location estimates. We have identified the need and lack of effective solutions without such information to assist monitoring in private indoor spaces without raising privacy concerns for the individuals involved.

In Chapter 4.7, we will introduce an infrastructure-free, privacy-preserving approach to semantic indoor localisation of users that aims to overcome those challenges by utilising smartwatch inertial data and ambient BLE signals sensed by the watch in the vicinity of the users monitored.

2.5 Human activity recognition

Human activity recognition (HAR), also referred to as activity detection or action recognition, is the automated process of identifying and categorising human actions or activities based on data collected from sensors or devices. HAR involves the application of machine learning and pattern recognition techniques to interpret and classify various physical activities or behaviours performed by individuals.

In line with our approach to human indoor positioning, our emphasis in the HAR domain remains on solutions that prioritise user privacy and uninterrupted conduct of their daily activities. We aim to provide recognition methods that do not impose restrictions on device usage or placement and are straightforward to deploy. To achieve this, we primarily concentrate on wearable and mobile solutions, particularly those based on smartphone and smartwatch inertial sensors. Nevertheless, we will also spotlight other noteworthy approaches that have demonstrated effectiveness in HAR.

2.5.1 Challenges in HAR

Identifying human activities is a challenging problem, largely because activity recordings, on which models are trained, have inherent challenges in their nature.

Data collection for HAR often involves multiple participants, leading to variations in the data even for the same activity due to the different styles the users perform or perceive an activity. This results in significant intraclass variability. At the same time, different activities can exhibit inherent similar data patterns, such as dusting and cleaning, leading to interclass variability. Class imbalance may also arise, since not all activities are performed in the same frequency during the

day; for instance, people typically brush their teeth only twice a day but walk for significantly longer periods [176].

Another challenge stems from the absence of a unified approach to annotating human activities. Unlike image data, activity data, particularly when recorded with inertial sensors, are not visually identifiable [177]. Consequently, the same semantic concept may encompass different underlying behavioural patterns, which can hinder model training and the generalisation of trained models; hence, mapping annotations across diverse datasets or domains becomes non-deterministic. Moreover, the continuous, composite, and concurrent nature of human activities makes consistent and accurate annotation challenging, often resulting in sparsely annotated or mislabeled data[176, 177].

The ubiquity of wearable sensors and IoT devices has led to users wearing multiple smart devices to enhance their daily lives. Often, users upgrade or switch to new electronic products that may share the same platforms (e.g., iPhone and Apple Watch). As a result, users anticipate that activity recognition systems will seamlessly adapt to these new devices and perform activity recognition as effectively as with their previous devices. However, different devices are typically positioned on various parts of the body (e.g., smartwatches on the wrist, smartphones in pockets), resulting in significant signal variations influenced by the movements of different body segments. Selecting the optimal device placement is a crucial consideration, as it can impact not only the range of activities the model can learn but also its ability to generalize across different placement scenarios[178].

2.5.2 HAR sensor modalities

Human Activity Recognition (HAR) employs various sensor modalities, including video cameras, wearable physiological and motion sensors, acoustic sensors [179], as well as device-free sensing methods like Wi-Fi [180]. Ambient sensors like infrared motion detectors and magnetic sensors have also been extensively used [181].

2.5.2.1 Vision-based sensors

Camera Camera-based HAR systems have traditionally focused on using monocular RGB videos, but recent developments have shifted focus towards 3D action recognition due to the availability of low-cost 3D capture devices like Kinect [32]. These involve estimating human activities by leveraging various information from skeleton joints, depth maps, etc. [33]. However, camera-based sensors have limitations, including user visibility, privacy concerns, and issues with light, clutter, and occlusion, affecting recognition performance.

Millimeter-wave (mmWave) mmWave technology operates in the frequency range of 30 GHz and 300 GHz. Due to their large bandwidth, these radars exhibit superior range resolution. The growing prevalence of low-cost, off-the-shelf radars has contributed to the popularity of mmWave-based sensing solutions. However, these devices, while cost-effective, come with resource constraints, presenting output in the form of point clouds rather than raw data. The variable number of points in each frame from mmWave radar captures introduces complexity in designing a neural network architecture capable of processing such data directly. Consequently, it is required to transform this data into other, more easy-to-handle formats, commonly images, to enable HAR from mmWave data recordings.

2.5.2.2 Ambient and object sensors

Wi-Fi WiFi-based human activity recognition relies on the concept that human movements and positions affect the propagation of signals travelling from the transmitter to the receiver. This includes both the direct and reflected propagation paths. By measuring the changes in the RSS or the CSI of the signal[182]

RFID RFID employs electromagnetic fields for the automatic identification and tracking of tags attached to objects, which store electronic information. There are two primary types of RFID tags: active and passive. RSS (Received Signal Strength) is a commonly used method for RFID-based activity recognition. It

operates on the principle that human movements can modulate the signal strength received by the RFID reader [34]. As such, it can estimate interactions with objects where RFID tags are attached, hence identifying low-level actions such as reaching, grasping and moving objects.

Environmental Various environmental sensors have also been explored for HAR, such as pressure, barometer, temperature and sound. These sensors are strategically placed in the environment to capture data related to changes and patterns in these environmental factors. By analysing the variations in sensor readings, machine learning algorithms can be employed to recognise specific human activities or behaviours. This approach enhances activity recognition by considering the contextual information provided by the surrounding environment. The integration of environmental sensor data contributes valuable contextual insights, making HAR more robust and capable of distinguishing between different activities based on the environmental conditions in which they occur.

2.5.2.3 Wearable sensors

Inertial sensors Wearable sensors on the other hand, such as accelerometers and gyroscopes, are commonly used for activity monitoring, mitigating privacy and security issues. However, they introduce their unique set of challenges, such as precise activity start and end time detection, device heterogeneities, and device-placement issues. Yet, their ubiquitous usage in everyday life has established them as the dominant sensor choice for the HAR problem.

Physiological sensors Electromyography (EMG) and electrocardiography (ECG) sensors have also been used in HAR. By measuring changes in muscular activity EMG sensors aim to identify fine-motor skill movements, or facial expressions. Similarly, the changes in heart rate encoded in ECG are utilised to identify the changes in activity patterns of the user [177].

2.5.3 HAR methods

Having summarised the most common sensors that can capture human motion patterns, in this section, we will present the research methodologies that have been employed to estimate the activities performed.

2.5.3.1 Vision-based approaches

Computer vision systems for human activity recognition employ various camera setups, including single cameras, multiple cameras, stereo vision, infrared or thermal cameras [39, 183, 184], with mmWave radars transformed to image-like data from the point-cloud space being used more recently [35, 36].

The increasing availability of depth cameras has allowed for interesting directions in vision-based HAR. In [185], a depth-based life logging system captures depth silhouettes, generates human skeletons, and employs hidden Markov models (HMM) for activity recognition, allowing for elderly monitoring and assistance.

While single cameras enhance deployability, multi-view setups offer detailed full-volume 3D data but face challenges like self-occlusion and silhouette quality [186]. The authors in [187] propose a video-based system for detecting abnormal activities in elderly home care, achieving a 95.8% recognition rate. Another approach [188] estimates joint angles from a 3D model, outperforming conventional methods in recognition rates, especially for complex activities.

Researchers address replicability and challenges on datasets that are recorded in more uncontrolled conditions. To achieve this, the authors in [189] propose a home activity summary system, considering unbalanced data and the presence of others. Moving towards a different direction, the work in [190] explored a scene-description-based approach, inferring human activities based on the observed changes in the user's surrounding environment. In this work, they introduce a probabilistic framework to capture uncertain knowledge in activity recognition, validated on real-world videos.

More recently, other point-cloud sensors, such as the mmWave radar have also been explored for HAR, as the sensor data from cameras carry a significant amount

of ambient information that may be of concern for privacy-sensitive applications. RadHAR [35]. mActivity [36] then improved on that approach

2.5.3.2 Interaction-based approaches

The concept of employing interaction-based ambient sensors for intelligent home automation has been an active area of research for many years. Pioneering projects such as GatorTech [191] and AwareHome [192] aimed to create living laboratories, developing proof-of-concept smart houses.

RFID technology has been widely explored for environment interaction detection [193–195], but it introduced additional burdens and electromagnetic exposures, prompting a shift towards low-power systems.

Towards a wireless sensing approach, the authors in [196–198] deployed a wireless sensor network (WSN) with 14 sensors in a real house, achieving high recognition accuracy in classifying seven activities using HMM, conditional random field (CRF) and a naive Bayes classifier (NBC).

Recent studies towards assisted living proposed hybrid models, combining artificial neural networks (ANN), support vector machines (SVM), and HMM for improved activity recognition performance [199]. In a similar direction, the authors in [200] proposed a unified framework for action prediction alongside activity recognition, achieving a 13.82% increase in accuracy using SVM-based kernel fusion [50].

2.5.3.3 IMU-based approaches

HAR based on IMU data has been an active area of research for almost two decades. Early works on traditional machine learning models for classification, e.g., SVM or random forests [201–203], achieve already significant results.

More recently, deep approaches have seen a surge of research activity to bypass the necessity of manually designing features. Convolutional modules as features extractors, oftentimes followed by RNN or LSTM layers, are a prevalent approach to deep HAR. Notably, [204] used deep neural networks for feature extractions before training on standard classification models. Similarly, [74, 75] built on this

idea, suggesting a combined approach between hand-engineered feature extraction and deep models. In particular, they encoded the information from raw data in a visual representation, e.g., a spectrogram, which was then used in a deep convolutional neural network to perform activity recognition. In a similar direction, wavelet transforms have also shown their effectiveness in recognising human activity patterns from various input signals. Example of input signals include video [205–208] and inertial measurement data [209–212]. In particular, discrete, complex, and continuous wavelet transform with the Symlet (Symlet 4) [205], Daubechies [206, 207], and Gabor [208] wavelets are employed to identify human activity from video frames accurately. Similarly, Biorthogonal (Bior2.2) [213], Haar [209], Mexican hat [212] and Daubechies (db10) [210] wavelets are used to recognise activities from inertial signals. However, the choice of wavelet and its effect on classification performance remains an open problem in these works.

A large part of HAR research pertains to the fusion of multiple sensor attachments. A plethora of approaches to sensor fusion have been explored that vary depending on whether fusion is attempted at the input stage (Early Fusion), if each sensor is modelled individually and then fused at the output stage (Sensor-based Fusion), or whether each axis of each sensor modality is further modelled individually (Axis-based and Shared-filter Fusion). Fusion is commonly achieved by concatenating the features in 1D or 2D matrices and then employing 1D or 2D autoencoders (dominantly, CNNs) to capture their dependencies within and across the various modalities. There are numerous works towards this direction: [214–218]. More recently, [215] also moved in the same direction, with the addition of estimating the IMU attachment's position in tandem. Finally, the work in [219] achieved very good performance from foot-attached IMU data trained on a convolutional neural network, augmented with a discriminator to generalise across different users.

While wearable and inertial sensors have seen a significant surge of research activity and effectiveness in HAR, most of these works focus on a combination of sensors attached/carried at various body parts, some of which are unnatural for everyday use. Interestingly, while a lot of the work has focused on smartphone-based

HAR or using a fusion of various sensor modalities, the usage of smartwatches as sensor modalities remains underexplored, despite their premium wrist-attached location. While smartphone-based models have effectively been used to tackle a range of interesting locomotive activities (walking, running, sitting, lying, etc.), there has not been extensive research regarding more complicated tasks. The placement of the smartwatch sensor on the wrist is thus promising to extend the scope of HAR models to include activities with high manual dexterity.

2.5.3.4 Hierarchical approaches

Most research works, as the ones discussed above, commonly address the problem of human activity recognition as a standard flat classification problem, i.e., by estimating a classifier that aims to identify each activity directly as is described. However, significant research also exists in hierarchical activity modelling. In this direction, activities are broken down to their individual components, and classification is performed by leveraging these together to provide a final classification estimate.

Bottom-up hierarchical activity recognition Bottom-up approaches focus on identifying fine-grained activity components before inferring high-level activities.

In bottom-up hierarchical learning, the authors in [220] introduced a two-stage hierarchical hidden Markov model (HMM) designed to sequentially classify user actions and activities. In their framework, actions were defined as short-duration movements, with five distinct categories: standing, walking, ascending stairs, descending stairs, and running. Once an action was identified, a longer temporal window was applied to analyse sequences of recognised actions for activity classification. The system classified user activities into three categories: shopping, taking a bus, and walking-based movement. Separate HMMs were trained to independently model action recognition and activity recognition processes.

Towards an approach of inferring activities by leveraging gestures, [67] introduced a model where gestures were initially classified at individual sensor nodes, followed by a central inference step that integrated these outputs with additional sensor data to determine the overall activity. Similarly, hierarchical hidden Markov models

(HHMMs) have been employed in HAR to detect latent variables representing subactions, enabling automatic optimisation within the model and outperforming standard hidden Markov models [221]. The authors in [222] proposed a two-stage recognition algorithm of high-level activities from gestures, based on dynamic time warping (DWT). In [223], a hierarchical Dirichlet process hidden Markov model was proposed to tackle trajectory-based HAR, a more complex recognition task. More recently, a deep encoder-decoder approach was proposed to automatically learn low-level activities from high-level ones across two levels of hierarchy [224]. Compared to other hierarchical methods, this work automates the hierarchy generation, which is useful to avoid manual work for the hierarchy design.

Top-down hierarchical activity recognition Hierarchical breakdown and feature selection approaches for various walking activities were also proposed in [40, 41, 66, 67], where hierarchical feature extraction and selection was accompanied by traditional machine learning classification, e.g., SVM or Naive Bayes classifiers, to separate active from static activities. Towards learning in multiple stages, the authors in [225] proposed a two-phase activity recognition system based on SVMs to address variability in smartphone sensor signals caused by differing device positions and orientations. In the first phase, gyroscope signals were used to determine the smartphone's placement, which was categorised into three groups: front trouser pocket, back trouser pocket, or a broader category encompassing the shirt pocket, backpack, messenger bag, or shoulder bag. Once the device position was classified, the second phase employed distinct activity classifiers tailored to each position type to recognise user activity.

Towards utilising both location and movement to enhance recognition of activities, the work in [226] presented a hierarchical activity detection model comprising a two-layer system. The first layer detected motion states and environmental context (e.g., presence in a coffee shop, restaurant, supermarket, or moving vehicle) using accelerometer and microphone data. The second layer refined this information to classify complex activities such as shopping, queuing, vacuuming, dishwashing, and

watching television. Decision trees and k-nearest neighbour (k-NN) were employed to recognise motion, environmental context, and specific activities.

A number of works in deep learning have explored hierarchical approaches to identify activities at various levels of granularity. The work in [227] proposed a hierarchical recognition approach employing two levels of continuous hidden Markov models (CHMMs). At the first level, CHMMs differentiated between stationary and dynamic activities. The second level further classified movements such as walking, ascending stairs, and descending stairs under dynamic activities, while distinguishing between sitting, standing, and lying down under stationary activities. In total, the system comprised eight CHMMs—two at the first level and six at the second—each trained on different feature subsets optimised for their respective classification tasks. Similarly, the authors in [228] employ a similar approach to first and second level classification, utilising linear discriminant analysis and artificial neural networks, and in [229], convolutional neural networks to classify abstract activity groups, followed by recognition of specific activities, while the authors in [40] employ deep neural networks to address the hierarchical learning task.

Activity recognition in ontology research Hierarchical activity modelling based on bottom-up hierarchies has also been an active area of research in the field of ontology. In [222], the authors utilised emerging patterns to identify high-level complex activities from gestures in real time, while [230] introduced an upper ontology tailored for smart environments, facilitating the formalisation of activity models for recognition tasks. Towards a different direction, [231, 232] utilise descriptions logics and temporal logic. In this context, descriptions logic captures the relationships between activities and their associated entities, while temporal activity modelling, characterises the connections between individual activities that constitute a composite activity. The authors in [233] demonstrated how Web Ontology Language (OWL) 2, with a focus on rule modelling, can be applied to both represent and reason about complex activities. The work in [234] presents a hybrid approach

to activity modelling, combining ontological knowledge engineering techniques with traditional data-driven classification models and active learning approaches.

A few works have tried to incorporate context-aware information into the activity recognition systems; [235, 236] proposed sensor ontologies for managing contextual data and developed activity ontologies to enhance modelling and recognition processes. In [237], the authors introduce the COSAR system to estimate activities by taking into account their probability of occurence within a given context. The approach combines statistical and ontological reasoning for activity recognition, leveraging ontology-based context modelling and derived semantic relationships to determine the feasibility of an activity within a given context. In a different approach to incorporating context, [238] combines ontological activity representation and semantic-based recognition, supported by an iterative process to enhance model accuracy and completeness and incorporate context-aware information through a service-oriented architecture, SOA. POLARIS [239], a real-time, unsupervised activity recognition system leverages an OWL 2 ontology to model activities and smart home infrastructure. It mines probabilistic semantic correlations between sensor events and activities, translating the ontological model into a Markov Logic Network for probabilistic and knowledge-based reasoning.

2.5.4 Summary and challenges

In this section, we have delved into the existing landscape of human activity recognition (HAR) methods. Our focus has been on understanding the sensor types most commonly employed in HAR, with particular emphasis on the prevalent use of inertial data sourced from sensors like accelerometers and gyroscopes. We have conducted a thorough exploration of the diverse range of methodologies that harness smartphone inertial data for HAR tasks, shedding light on the types of features and models that are frequently employed in these approaches.

While smartphone-based HAR solutions offer great promise, we have also acknowledged certain limitations, particularly regarding their ability to provide continuous tracking and their effectiveness in recognising a wide spectrum of human

activities. Furthermore, we have recognised that the context in which activity recognition is required can vary significantly, and different mobile applications may demand varying levels of granularity in activity information. Consequently, we have underscored the need for centralised systems capable of recognising activities across a range of contexts and with varying degrees of detail.

In Chapter 5.7, we will delve into the utilisation of hierarchical structures to design deep learning architectures, to achieve a semantic understanding of human activities over several granularity levels. These architectures aim to achieve accurate and efficient recognition of activities of daily living (ADLs) using data from a single smartwatch.

2.6 Generalisation and learning

Machine learning models are traditionally trained under the closed-world assumption, wherein all data in the training dataset are independently sampled and identically distributed, meaning they originate from the same underlying distribution. At the same time, each model is constrained by a specific scope dictated by the data in its training dataset. In other words, a model can only proficiently predict incoming data from the classes it has been trained to recognise.

However, these assumptions may not hold when these models are deployed in real-world scenarios. Instead, incoming data may showcase different styles within existing classes (covariate shift) or introduce new concepts (semantic shift). Models need to adeptly capture such inconsistencies in incoming data, highlighting previously unseen data and, whenever possible, discerning the underlying patterns of the new data.

This thesis poses the above two problems under the scope of data explainability for human activity recognition. While an overall common understanding exists of what actions are expected when an activity is performed, the exact specifics are not necessarily identical among different users. Instead, the meaning of an activity label is tightly coupled to each user's task perception. For example, one user may consider the action of "preparing a sandwich" as "cooking", while another might assign this label only to the preparation of more complicated meals.

Such inconsistencies become particularly prevalent when considering examples coming from different datasets. Particularly when these are crowdsourced, the users themselves are responsible for their annotation assignment, inducing noise arising from their diverse task understanding, from the user's annotation consistency and inadvertent annotation errors. As a result, when investigating two datasets with similar semantics using the same model, it is likely to face unexpected obstacles in model *generalisation*, even for data of identical semantics, when the underlying patterns that govern each activity class vary. In such scenarios, *learning* semantic mappings of incoming data to the knowledge embedded in the model's training dataset is critical to explaining and understanding the task perception governing each activity class in a new dataset. Due to the *noise* incurred from such annotations inconsistencies, studying the effect of label noise in learning is an essential factor in establishing robust model training and generalisation capabilities.

As such, this chapter will focus on three main directions that we believe are key to delivering explainability mechanisms for incoming activity data: model generalisation, which might be hindered when inconsistencies exist; transfer learning techniques, that can be utilised to reveal semantic nuances across datasets; and learning under label noise, which is prevalent when data collection is performed in-the-wild. Most of these research directions have been explored for image data. However, our work in this thesis focuses on inertial sensor data from mobile, wearable devices; hence, we will attempt to include all relevant work in this field.

2.6.1 Model generalisation

Generalisation is the ability of a trained machine learning model to perform well on new, unseen data that were not part of the training set. Incoming data may be similar to those the model has been trained on or have significant variations in their distribution or semantic concepts. Samples that only face a slight covariate shift with respect to the training samples are known as In-Distribution (ID). In contrast, samples with large covariate and semantic shifts are known as Out-Of-Distribution (OOD).

ID generalisation is a critical component of standard model training: models are trained not to overfit the training dataset. This is achieved through regularisation mechanisms, introduced in Section 2.2.3. In this section, we will focus on OOD generalisation. Ensuring both ID and OOD generalisation is imperative for a model's robust performance. While ID generalisation guarantees the model's ability to make accurate predictions for data originating from the same distribution as the training dataset, OOD generalisation focuses on identifying samples that deviate from this constraint, detecting instances with distributional or semantic content not encountered during model training.

The process of OOD detection encompasses various applications, including outlier detection, aimed at identifying samples significantly differing from other observations; anomaly detection, when pre-existing, well-defined rules govern the characteristics of each class; novelty detection, to identify samples seemingly originating from entirely unseen classes; and open set recognition, that aims to combine these notions by correctly classifying relevant samples in known classes while actively rejecting samples from unknown classes [240].

Regardless of the exact application, failure to recognise an OOD sample and consequently assign that sample to an ID class significantly compromises the reliability of a model. Hence, we will discuss the principles and methodologies of OOD detection regardless of the exact application. These are categorised based on the primary mechanism employed to achieve OOD generalisation: classification model, probability density, feature distance and data reconstruction.

2.6.1.1 Classification-based OOD detection

Classification-based methods form the large majority of OOD detection methods, aiming for a classifier's output to distinguish between known and unknown samples robustly. This is often considered as estimating model uncertainty. In this front, MC-dropout [241] accurately estimates a model's predictive uncertainty for models trained with Dropout regularisation, allowing for ID/OOD detection by thresholding the output confidence score. Deep Ensembles [242] is also a

notable approach that combines the predictions from N independently trained models, while Prior Networks [243, 244] explicitly learn a sharp/flat distribution for ID/OOD data, respectively.

Another common approach aims to amplify the distance between the distributions of ID and OOD samples. ODIN [245] notably uses temperature scaling and input perturbation to identify ID and OOD samples: scaling the model's outputs (logits) with a temperature scaling parameter T before applying the softmax function increases the separability between the output softmax probability distributions for ID and OOD samples. This is further ameliorated when small perturbations ϵ are added to the input samples. G-ODIN [246] redefines ODIN using the notion of decomposed confidence, which directly correlates logits to the domain posterior probability by learning a dividend/divisor scoring function. DeepSVDD [247] maps ID examples into a hypersphere so that the description of ID samples is bounded, and deviations from this representation are OOD.

ReAct [248] and NMD [249] demonstrate the effectiveness of utilising the outputs of activation functions in both logits and latent space to increase separability between ID and OOD samples and decide on OODness based on the distance of a sample to the latent activation means of the training batches. Towards modelling each ID class independently, OpenMax [250] notably introduces a new decision layer that substitutes the SoftMax function with a set of per-class probabilistic models, such that samples from unknown classes can be directly discarded. In the same direction, M2IOSR [251] borrows from [252], where the authors demonstrated the effectiveness of transformers in OOD detection. It utilises an encoder that maximises the mutual information between sample inputs and their latent features, which are constrained to class-conditional Gaussian distributions through a KL-divergence loss function.

Finally, GradNorm [253] introduces the vector norm of gradients as an OOD scoring function instead of latent feature representations, using the gradient of the Kullback-Leibler (KL) divergence between the softmax output and a uniform distribution; ID samples are expected to have larger KL divergence, as we expect a peaked distribution around the samples corresponding class.

In an attempt to simultaneously recognise and learn new classes, the authors in [254, 255] introduce group-based and hierarchical-based frameworks, respectively. While MOS [254] groups classes to semantically similar groups and introduces an `other` category in each group for OOD samples, the authors in [255] directly identify ID and learn OOD samples at different levels of hierarchy. Starting with a predefined hierarchy of classes, the model assigns input samples to the appropriate node in each level of the hierarchy or creates a novel class at that level based on the prediction's confidence. Confidence is defined as the KL divergence of the output softmax probabilities to the uniform distribution.

While most classification-based methods for OOD detection rely on handling output or latent feature representations, a few methods make a decision based on the model's energy or entropy. The authors in [256] defined the energy of a model as the denominator of the softmax function and classified samples as ID or OOD by thresholding the above energy function. They also demonstrated that exposure to OOD samples during training helps create an energy gap by assigning lower energies to the in-distribution data and higher energies to the OOD data, further improving OOD detection. JointEnergy [257] redefines the concept of energy as the summation of energy scores across all classes to improve [256] in the multi-class set-up. Viewing the problem from an entropy perspective, the authors in [258–260] encouraged a high-entropic prediction (close to a uniform distribution) on exposure to OOD samples.

2.6.1.2 Density-based OOD detection

Density-based methods model the ID samples with some probabilistic model. Incoming data that are found to be in low-density regions are considered OOD. In [261], the authors fit a multivariate Gaussian distribution, then decide on OODness using the Mahalanobis distance between an incoming sample and the expectation of the training ID samples. DAGMM [262] employs a deep autoencoder and then fits the encoded latent representation using Gaussian Mixture Models (GMMs). In a similar direction, the authors in [263] also utilise an autoencoder to

extract robust latent representations and jointly learn an autoregressive density estimator using maximum likelihood on the latent features. Working towards a smoother interpretation of OODness, where semantic shift is considered the primary OOD factor, FS-OOD [264] also employs GMMs to model the output and latent feature representations from a deep network and utilises the learned models to define SEM, a scoring function to distinguish between ID and OOD.

2.6.1.3 Distance-based OOD detection

Distance-based OOD detection methods decide on OODness with respect to the distance of incoming samples to the centroids of known classes.

Popular work MD [265] models latent features with a class-conditional Gaussian distribution as a substitute for the softmax function, then uses the Mahalanobis distance with respect to the closest class-conditional distribution. Each class distribution is defined using the class means and covariance of the training samples. RMD [266] further improves the Mahalanobis distance approach by modelling the training data both in a class-conditional and a class-agnostic way and calculating the difference in the MD scores for an input sample from the two distributions, improving OOD detection for near-OOD samples.

Skewing away from the Gaussian assumption, the authors in [267] employ RBF networks to simultaneously classify data and learn a centroid-based representation for each class and estimate uncertainty as the distance of an input sample to the closest class centroid. In a different cluster-based approach [268], the authors jointly learn a deep classification model and a latent feature clustering on the data, filtering out OOD samples that cannot robustly be assigned in the learned clusters. In [269], the authors explore a nearest-neighbours approach to identify OOD samples. Using the model's normalised logits as a latent feature representation, a sample is considered as OOD by thresholding on the k^{th} nearest neighbour distance between its embedding and the training set embeddings.

2.6.1.4 Reconstruction-based OOD detection

Reconstruction-based OOD inspects the output of an encoder-decoder pair trained on the training ID samples. Most reconstruction-based methods use an autoencoder or variational autoencoder to perform the reconstruction. Notably, [270] uses a CNN encoder-decoder pair to learn the concepts existing in the training samples and then uses its reconstructed output as input to a discriminator model. C2AE [271] further improves on the above work using class-conditional autoencoders (instead of class-agnostic as in [270]). It enforces conditional reconstruction constraints, allowing the model to learn approximate distributions of reconstruction errors for both ID and OOD samples. On a similar approach, the authors in [272] also explore class-conditional variational autoencoders (VAEs) forced to approximate different multivariate Gaussian models in conjunction with a probabilistic ladder network that allows for the representation of long-term dependencies in the VAE. The authors in [273] also long-term dependencies by combining a VAE and LSTM module to capture data correlations over shorter and longer periods, effectively identifying OOD events when they occur.

In [274], the authors introduce GANs as an alternative to AEs/VAEs, classifying samples based on the reconstruction errors, while [275] employed a similar approach using an ensemble of GAN models, further improving OOD generalisation.

Model generalisation identifies previously unseen samples from a trained model, hinting when the model is uncertain of its predictions. The following section will explore how the model can learn to further adapt to these new concepts.

2.6.2 Transfer learning

In the previous section, we highlighted existing works that allow models to identify inconsistencies in incoming samples with respect to the training data. In this section, we will explore mechanisms that would enable adapting a model's predictive capabilities by leveraging knowledge from a related domain (called source domain)

to improve the learning performance or minimise the number of labelled examples required in a target domain.

Depending on the application focus, transfer learning can be viewed under different categorisations [276]. When considering primarily the label information across source and target domains, transfer learning problems can be divided into three categories: transductive, when labels exist in the source domain but are lacking in the target domain; inductive, when labels are available in both source and target domains; and unsupervised, when no label information is available in any domain. When the feature space is also of interest, we define the approaches as homogeneous when the source and target feature and label spaces are identical and heterogeneous when the source and target feature and/or label spaces are distinct. The special heterogeneous case when both the feature spaces and label spaces differ is often known as cross-modality transfer learning [277]. Finally, transfer learning approaches can be categorised into four groups based on the mechanisms employed [277]: instance-based, when directly reusing samples from the source domain; feature-based, when the original features are transformed to create a new feature representation; parameter-based, when model tuning is involved; and relational-based, when the relationship or rules learned in the source domain are transferred to the target domain.

Here, we will explore the methodologies based on the latter categorisation, as we consider it to encompass all the above adequately. We will, however, highlight other categorisations where appropriate.

2.6.2.1 Instance-based transfer learning

Instance-based approaches are based on transferring the knowledge in the source domain to the target domain by re-weighting or re-sampling (a subset of) instances from the source domain.

Notably, TrAdaBoost [278] transferred pioneering work AdaBoost [279] to the transfer learning domain. Using a small subset of accurately labelled data from the target domain, TrAdaBoost evaluates each sample in the source domain and identifies

instances with low scores as coming from a different distribution. It then utilises all data from the source and target domains to train a new model for the target task by appropriately leveraging each sample's weights by decreasing/increasing the weights of correctly/incorrectly estimated samples, respectively. The authors in [280] further advanced TrAdaBoost to leverage knowledge from multiple source domains simultaneously. More recently, the authors in [281] further advanced this idea by relaxing the requirement of source-domain data, using pre-trained models as feature extractors for the target domain instead. At the same time, GapBoost [282] identified the performance gap as a robust metric for the divergence between source and target domains and used this metric as a proxy to identify the appropriate instance weights directly.

2.6.2.2 Feature-based transfer learning

Feature-based methods aim to minimise the distribution mismatch between the source and target domains by transferring or transforming features to another space [283]. This can be either a new, domain-invariant space or a combination of the source and target domains [277].

In this context, DaNN [284] first introduced Maximum Mean Discrepancy (MMD) to reduce the distribution mismatch between two hidden layer representations induced by samples drawn from different domains. In this way, hidden layer representations are encouraged to be invariant across different domains. DAH [285] then utilised multi-kernel MMD (MK-MMD) to relieve the task-specialisation in the near-output layers and close the distribution mismatch between the source domain and the target domain by leveraging the latent representations over several hidden layers. CORAL [286] and Deep CORAL [287] also offered a simpler implementation of this approach by using the distance between the second-order covariances of the source and the target features as training loss.

More recently, GAN-based approaches have been introduced to make the distributions of source and target domains more distinguishable. JAN [288] follows a similar idea to MK-MMD by introducing Joint MMD (JMMD), which measures

the discrepancy in the joint distribution of all latent representations. The authors demonstrate that further enhancing this loss with adversarial training can better highlight any domain shift between source and target data. DANN [289] also employed adversarial training by introducing a simple gradient reversal layer that allows the learning of robust hidden representations that are, however, domain-invariant.

Feature-based unsupervised transfer learning primarily focuses on clustering approaches. Indicatively, CoCC [290] addresses the problem for document classification analysis by jointly clustering document-to-word representations from the source and target domains, aiming to minimise the loss in mutual information. STC [291] further improved on the above result by relaxing the label requirement by identifying a shared feature cluster space between a source and target dataset.

2.6.2.3 Parameter-based transfer learning

Parameter-based transfer learning utilises parts of a model trained in the source domain in the target domain. This can be a combination of freezing, finetuning or adding new layers in the learned model to allow further model capacity to extend to the knowledge embedded in the target domain. DAM [292], fastDAM, and UniverDam [293] directly amend the training objective function to allow for parameter learning in the transfer domain, incorporating information from the labelled target subspace and pre-trained classifiers in the source domain.

In an approach towards parameter sharing, MMKD [294] identifies the subset of parameters that contain meaningful information for the target domain (e.g., the set of parameters that correspond to a subset of classes of interest) using Least Square Support Vector Machines (LS-SVM). TransEMDT [295] moves towards finetuning a model's parameters and combines decision trees with K-Means clustering by incrementally adapting the model's parameters with labelled target data. In a different direction, MRN [296] imposes the tensor normal distribution as a prior to model parameters. The authors demonstrate that this approach relieves the task-specific knowledge learned towards a classifier's output layer, instead jointly learning transferable features and multilinear relationships.

Finally, LWE [297] demonstrates the capabilities of employing model ensembles for the transfer learning task, taking advantage of the different predictive powers of several models trained on different source domains or using different learning algorithms, transferring the combined knowledge to a new, target domain.

2.6.2.4 Relational-based transfer learning

Relational-based transfer learning approaches are not as common as the methodologies mentioned above, even though they do not assume the source and target data to be independent and identically distributed (i.i.d) and would thus be more widely applicable. Based on the assumption that similar relations exist in different domains, most existing algorithms are based on statistical learning techniques.

In this direction, TAMAR [298] transfers relational knowledge across domains using Markov Logic Networks (MLNs). The information embedded in the MLNs from a source domain is then leveraged to assist MLN learning in a target domain. The algorithm comprises two stages: constructing a mapping from a source MLN to the target domain with a weighted pseudo-log-likelihood measure (WPLL) and revising the mapped structure in the target domain. The revised MLN serves as a relational model for inference in the target domain. Further extending the idea, the authors in [299] proposed a method for transferring relational knowledge based on second-order Markov logic, discovering structural regularities in the source domain instantiated with predicates from the target domain.

Transfer learning techniques allow for the adaptation of a model's predictive capabilities from a source domain to a *stationary* target domain. However, it relies on consistent data distributions in the target domain, which might not be the case when models are deployed in the real world. In reality, incoming data may change over time, be inconsistent in their perception or, when available, have incorrect annotations. In the next section, we will explore how to effectively train models with such noisy data to ensure robust learning and performance.

2.6.3 Learning from noisy labels

In the previous sections, we have demonstrated popular methods that identify unforeseen input and transfer acquired knowledge to identify tasks in a new domain. However, these methods do not account for the inherent imperfections in labelled datasets. Mislabelled instances can complicate the training process, leading the model to overfit the noisy samples, hence hindering model performance and robust generalisation. In this section, we will investigate mechanisms that establish robust learning in the presence of noisy labels.

Approaches to handling noisy labels can be categorised based on the primary mechanism employed to address the label inconsistencies. These can be architecture-based, when the model architecture itself is adapted to capture label noise directly; regularisation-based, when standard techniques to avoid overfitting are employed; loss-correction-based, when a loss function is designed explicitly, or label robustness is added in the optimisation process; and, sample-selection-based, when the noisy subset of a dataset is identified and removed ahead of the training process.

There are two types of label noise that we will explore in this section. Label-independent noise, when corruption may occur at any sample irrespective of its given annotation, that can modelled by a noise transition matrix T; and, label-dependent noise, when the nature of some samples may lead to higher levels of confusion and hence untrustworthy annotations, modelled by more complex representations.

2.6.3.1 Architecture-based learning

A radical approach to learning in the presence of noisy examples employs the design of noise-handling architectures. Yielding minor modifications to existing architectures, a common approach to handle label-independent noise is introducing a noise adaptation layer following the output softmax function that represents the noise transition matrix T. The parameters of the noise adaptation layer are learned during training time, then at inference time, the layer is removed to obtain the model's prediction. In this direction, the Dropout [300] method has demonstrated superior classification performance but significantly increases training time. In a

different approach, the authors in[301] combine the noise estimation and outlier detection problems in the noise adaptation layer. In this way, they jointly learn the label noise in the layer's parameters and create an outlier class to identify samples that have been erroneously assigned one of the known annotations but are, in reality, out-of-distribution. Similarly, the work in [302] learns the noise distribution in the noise adaptation layer as a function of both the true label and the input features when this dependency is desired and trains the model with standard EM.

To capture more complex noise distributions, a few dedicated architectures have been proposed. The authors in [303] employ a probabilistic graphical model that only requires a small amount of clean data for training and can be represented as a set of two CNN modules, hence seamlessly embed to other deep architectures. For each sample, the model captures the true labels for other similar samples and estimates how likely it is for the sample to be mislabeled. Masking [304] introduces a human prior for commonly confused categories, denoted as *structure*, which is modelled using GANs. This notion constrains the noise transition probability matrix, hence significantly relieving the computational cost. Finally, RoG [305] also utilises generative models to learn label noise directly from latent features generated by pre-trained models without the need for model retraining or modifications to the model itself.

2.6.3.2 Regularisation-based learning

Regularisation approaches are also often employed to avoid overfitting, which makes the model more resilient to label noise. Standard regularisation approaches, such as dropout [300] and batch normalisation [306] can be employed and improve performance, but more sophisticated approaches have been developed to address the label noise problem.

A number of methods demonstrate the effectiveness of prior knowledge embedded in pre-trained models. Pretraining [307] substantiates, through empirical evidence, that fine-tuning on a pre-trained model significantly enhances robustness compared to models trained from scratch. The universal representations acquired during

pretraining act as a safeguard, preventing model parameters from being updated erroneously by noisy labels. Similar to the transfer learning approaches mentioned in Section 2.6.2, robust early learning [308] involves the classification of critical and noncritical parameters for fitting clean and noise labels, respectively. A distinct update rule penalises only the noncritical parameters, thereby mitigating the impact of noisy labels. Towards constraining the parameters corresponding to noisy samples, label smoothing [309] estimates the marginalised effect of label noise during training, reducing overfitting by preventing the DNN from assigning a full probability to noisy training examples. Instead of employing a one-hot label, the noisy label is combined with a uniform mixture over all possible labels.

Bilevel learning [310] introduces a novel bilevel optimisation approach. Contrary to conventional methods, this approach incorporates a regularisation constraint as an optimisation problem. It effectively controls overfitting by dynamically adjusting weights on each mini-batch, optimizing them to minimize errors on a small validation dataset. PHuber [311] proposes a composite loss-based gradient clipping, a variation of standard gradient clipping tailored for label noise robustness.

Annotator confusion [312] moves towards an ensemble-like approach, assuming the presence of multiple annotators. It introduces a regularised EM-based approach to model label transition probabilities that facilitates the convergence of estimated transition probabilities to the true confusion matrix of annotators.

Finally, adversarial training approaches have also proved successful. Pioneering work [313] enhances noise tolerance by encouraging a Deep Neural Network (DNN) to correctly classify both original inputs and hostilely perturbed ones, improving the model's resilience to noise.

2.6.3.3 Loss-correction-based learning

The categorical cross entropy (CCE) loss is widely employed for classification tasks due to its rapid convergence and high generalisation capabilities. As such, most selected loss functions for noisy learning are based on the CCE. The authors in [314] demonstrated that in the presence of noisy labels, the robust mean absolute error

(MAE) loss is a more robust choice because it satisfies a noise-tolerant condition. Despite its effectiveness, MAE loss struggles with complex data. GCE [315] further advanced this approach, aiming to combine the advantages of both MAE and CCE losses. Inspired by the symmetricity of the Kullback–Leibler divergence, the symmetric cross entropy (SCE) [316] combines a noise tolerance term, reverse cross-entropy loss, with the standard CCE loss.

Towards adjusting existing loss functions rather than defining new ones, the authors in [317] proposed the backward and forward correction methods; in the backward case, model retraining is performed after the noise transition matrix is estimated from the softmax output to incorporate the learned label noise. In contrast, the forward correction approach employs a linear combination of a DNN's softmax outputs before applying the loss function, multiplying the estimated transition probability with the softmax outputs during forward propagation. Gold loss correction [318] relies on clean validation data or anchor points for loss correction, resulting in a more accurate transition matrix that enhances correction robustness. T-revision [319] provides a solution to infer the transition matrix without anchor points, and dual-T [320] factorises the matrix into two easy-to-estimate matrices, avoiding direct estimation of the noisy class posterior.

Bootstrapping [321] introduces the concept of label refurbishment to update the target label of training examples, creating a more coherent network capable of evaluating the consistency of noisy labels. Dynamic bootstrapping [322] adjusts the confidence of individual training examples dynamically. Self-adaptive training [323] uses exponential moving averages to alleviate instability, considering local intrinsic dimensionality to evaluate confidence. SELFIE [324] introduces refurbishable examples that can be corrected with high precision, focusing on examples with consistent label predictions.

2.6.3.4 Sample-selection-based learning

In sample selection methods, the trustable part of the dataset is identified for robust model training. The remaining samples are either discarded or refurbished

to their predicted true label.

The authors in [325] advocate towards separating the decision of when to update from how to update. This involves maintaining and updating two DNNs simultaneously, selecting examples based on their disagreement. The "small-loss trick" is another selection criterion, treating training examples with small losses as potentially true-labelled examples due to the memorisation effect of DNNs. MentorNet [326] employs a pre-trained mentor network to guide a student network collaboratively, using the small-loss trick for correct label predictions. Co-teaching [327] and Co-teaching+ [328] maintain two DNNs, each selecting small-loss examples and training their peer DNN. JoCoR [329] reduces network diversity via co-regularisation to make predictions closer.

ITLM [330] iteratively minimises trimmed loss by alternating between selecting true-labelled examples and retraining the DNN. INCV [331] divides noisy training data, uses cross-validation for classification, and employs co-teaching for DNN training. O2U-Net [332] repeats the training process with cyclical learning rates, retraining the DNN from scratch for clean data after false-labelled examples are detected.

To address the limitation of discarding unselected training examples, SELF [333] combines with a semi-supervised learning approach to progressively filter out false-labelled examples, leveraging the mean-teacher as a running average model. DivideMix [334] transforms noisy data into labelled and unlabelled sets using Gaussian mixture models and applies MixMatch. SELFIE [324] is a hybrid approach combining sample selection and loss correction, considering more training examples for DNN updating.

2.6.4 Summary and challenges

In this section, we have explored some important challenges that can manifest when deploying models in real-world scenarios. In such operational scenarios, a model may be presented with incoming samples that belong to previously unseen categories; necessitating robust out-of-distribution detection mechanisms is key for the model

to identify such examples and communicate its uncertainty to the user. In a different direction, models may encounter the need to operate in novel scenarios of a similar nature, demanding adaptability through transfer learning. In both these scenarios, incoming samples may be afflicted with label noise, hence training a model to effectively handle such noise is an important part of real-world model deployment.

The research area of out-of-distribution detection has thoroughly studied the problem of handling previously unseen incoming data. Most works, however, assume that samples deviate from the model's embedded knowledge simply when they are sourced from a new dataset, even when their annotation matches the learned classes. This contradicts the intuitive human understanding of data similarity. In reality, non-expert users would expect a mobile application to confidently capture their performed activities as they perceive them. This principle also extends to samples that exhibit some dissimilarity with the learned classes but share significant semantic information; for instance, "sitting" and "watching TV" are highly correlated, given that people commonly watch TV while seated.

However, the limited research addressing out-of-distribution detection from a semantically coherent perspective, particularly in the context of near-OOD detection, faces challenges in establishing such logical connections. While it is reasonable to anticipate that *a new dataset may not encompass precisely the same class set as a model's training dataset*, current methodologies only permit for activities not strictly belonging to the learned classes to be categorised as (near or far) OOD, which is unnecessarily impractical for everyday applications. Moreover, when data from known classes are erroneously classified as OOD, there lacks a systematic methodology to discern the subtle nuances in the data hindering the model's generalisation. Although the domain of transfer learning has demonstrated mechanisms for extending a model's capacity to learn from *new samples introduced by users with different task understandings*, it fails to offer a comprehensive explanation for the necessity of such adaptations. At the same time, while adaptation to accommodate knowledge transfer across different users, devices or environments exists, we have not identified any works that study the realistic problem of *data being*

annotated at different time frequencies. In the realm of learning from noisy data, efforts have been made to address the *uncertainty arising from label noise*, stemming both from differences in underlying data patterns and erroneous annotations. However, the estimation of noise type typically adopts a numerical perspective rather than providing a semantic, explainable understanding through the model's lens, a viewpoint we consider valuable for real-world model deployment.

In Chapter 6.7, we will investigate this problem in the context of HAR. Specifically, we will focus on discerning and alleviating the various causes that may impede a model's generalisation or the seamless transfer of knowledge to a new target dataset comprised specifically of ID and near-OOD samples. Our approach aims to bridge the disparities across the datasets, bringing to light both the semantic similarities and discrepancies (noise) across corresponding classes in the two datasets. This strategic approach paves the way for robust generalisation and effective transfer learning across similar domains.

2.7 Conclusions

In this chapter, we have discussed the relevant research pertaining the key research directions explored in this thesis: identifying human location in indoor spaces, estimating information about the activity performed, and handling real-world data. We have explored existing datasets and highlighted the advantages and potential challenges that arise from the current state-of-the-art methods, which we will aim to address in the following chapters of this thesis. In particular:

In Section 2.3, we summarised existing datasets in the areas of human indoor positioning and human activity recognition. We have highlighted the lack of available datasets that contain smartwatch sensor recordings, particularly alongside ambient signals and location information, which would enable usability of datasets in the positioning scenario. In Chapter 3.4, we address the lack of such data by introducing two smartwatch-based datasets that we have collected and released with this thesis: **WatchBLoc**, for the positioning scenario, and **watchHAR** for HAR problems.

In Section 2.4.3, we provided an overview of various existing indoor positioning methods, covering visual, radio-based, and inertial positioning techniques. We have considered sensor types, environmental constraints, and the challenges of positioning individuals in private indoor spaces. We have stressed the importance of privacy-respecting methods that steer clear of fingerprinting and extensive calibration and have recommended the use of ambient sensor signals that do not require additional infrastructure. While public indoor spaces have been extensively studied, solutions for private environments are limited, making WiFi and Bluetooth Low Energy (BLE) approaches with smartphones more prominent. Recognising the complexity and costliness of many solutions, we have highlighted the need to shift to wearable sensors, such as smartphones and smartwatches, emphasising their lightweight nature and accessibility. Smartwatches emerge as a promising alternative, addressing challenges in continuous monitoring within private indoor spaces, but remain understudied in literature.

In Chapter 4.7 we will move towards this direction, introducing an infrastructure-free, privacy-preserving approach that uses smartwatch inertial data and ambient BLE signals for semantic indoor localisation in private spaces.

In Section 2.5.3.4, we discussed research directions towards human activity recognition, focusing on understanding the prevalent sensor types, particularly the common use of inertial data from sensors like accelerometers and gyroscopes. We have summarised various methodologies leveraging smartphone inertial data for HAR, highlighting the frequently employed features and models. While smartphone-based HAR solutions show promise, we have identified certain limitations, especially in continuous tracking and in recognising a broad spectrum of human activities. Additionally, we identified the varying contexts in which activity recognition is needed, with different mobile applications requiring different levels of granularity in activity information. Hence, we emphasise the necessity for centralised systems capable of recognising activities across various contexts and detail levels.

In Chapter 5.7, we will explore a centralised approach to activity recognition that uses hierarchical structures that achieve a semantic understanding of human activities across multiple granularity levels.

In Section 2.6.4, we explored significant challenges that may arise during the deployment of models in real-world scenarios. We have highlighted the need for robust out-of-distribution detection mechanisms that are capable of identifying incoming samples from previously unseen categories and communicating the uncertainty of predictions to the user. Additionally, models may need to operate in novel scenarios of a similar nature, requiring adaptability through transfer learning while effectively handling incoming label noise. In these directions, we have identified a gap in solutions that do not only discern these challenges but also provide explainable information regarding the incoming data. Such an approach would significantly assist in understanding the varying task perceptions of human activities from different users and set the basis for robust generalisation, effective transfer learning and continual learning across similar domains that are currently commonly considered out-of-distribution despite their semantic similarities with the in-distribution domain.

In Chapter 6.7, we will delve into the problem of HAR, specifically focusing on discerning and alleviating the causes that may impede a model's generalization or the seamless transfer of knowledge to a new target dataset samples from semantically similar, albeit not identical classes.

Chapter 3

Contributed Datasets

3.1 Introduction

In this chapter we will introduce **WatchBLoc** and **watchHAR**, two smartwatch-based datasets that we collected during this thesis, and have released to the research community through the Zenodo platform.

Ethics protocol Both datasets involve human participants and log personal and identifiable data, including camera recordings (capturing facial features), inertial data (from which gait estimates—an identifiable trait—can be extracted), and ground truth information about participants' home environments.

To safeguard participant privacy and protect personal data, a full CUREC ethical review was conducted on the data collection protocols, covering participant recruitment, informed consent, data anonymisation, and secure data storage and processing. The proposal received approval under reference SSD/CUREC1A CS_C1A_19_012, encompassing the design, recruitment, and data collection for both datasets. The following protocol was followed:

- A call for participants was distributed via the Cyber Physical Systems Group mailing list, with recipients permitted to share it further. There were no

exclusion criteria; participation was open to all interested individuals.

- Data collection in participants' homes when this was necessary was conducted without video recordings to protect privacy. Video recordings were only made in a controlled laboratory setting for ground truth verification during data analysis and were not included in the released datasets.

- Data were collected using smartwatch sensors via tailored recording applications and transferred manually to secure desktops.

- Access to identifiable user data was restricted to research group members. Any directly identifiable information (e.g., names, video recordings) was permanently deleted following data analysis and anonymisation.

3.2 The WatchBLoc dataset

In this section, we will present **WatchBLoc**, a new dataset we collected to achieve indoor localisation for home environments using BLE RSSIs from ambient IoT signals and smartwatch IMUs.The dataset is publicly available at:

`https://doi.org/10.5281/zenodo.7039554`

3.2.1 Motivation

WatchBLoc is a dataset comprising ambient BLE, inertial data and ground truth room-level location for training and evaluating human indoor positioning models, particularly in scenarios where fingerprinting, environmental maps, and other location references are unavailable due to privacy constraints. While existing indoor localisation datasets primarily rely on sensor modalities that may not align with the privacy and cost restrictions of personal indoor spaces, there is a notable absence of datasets that provide both ambient signals and room-level ground truth location information.

Moreover, most available datasets originate from controlled studies that impose constraints on participants' behavioural patterns and limit the number of accessible

spaces, often to assuming locations within the same room correspond to different "rooms". Such restrictions fail to capture the diversity of real-world user behaviour as they move within a space performing their typical daily activities.

To address this gap, WatchBLoc offers a dataset designed to support the development of robust and accurate learning models for inertial tracking in realistic indoor environments. The goal is to create a dataset that can provide accurate location information about the users occupying the space, while being able to maintain their naturalistic behaviour, without relying on heavy, cost-prohibitive, or intrusive infrastructure.

3.2.2 Dataset description

The dataset was collected over two environments, utilising a smartwatch for the collection of inertial data, and BLE beacons positioned strategically across the rooms of the environments to act as ambient sensor signals. A recording application, installed on the smartwatch, was responsible for recording both ambient signals sensed due to the BLE beacons, and inertial data in the smartwatch's IMU. A separate application was also installed, for the users to record their ground truth room transitions when changing location.

Environment settings To reflect sensor readings under everyday usage, the data were collected over two environments; a *demo-home*, i.e., an office space set up to facilitate the experiment, and a *real-home*, i.e., a standard flat. The floorplans of the data collection environments are shown in Fig. 3.1.

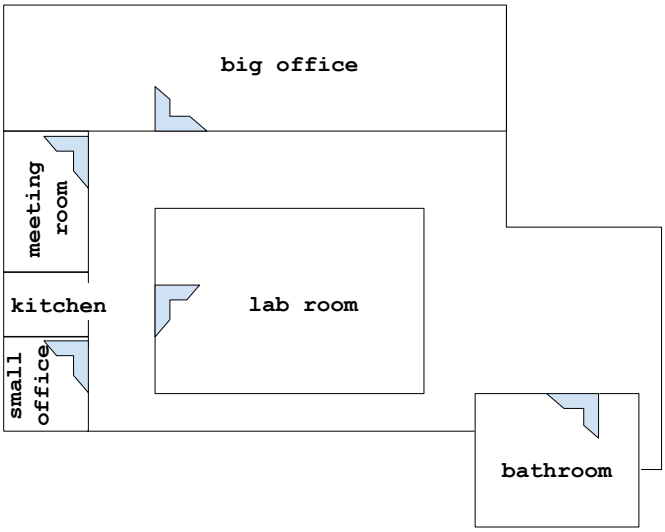
The choice of environments was practical as well as informational; while the *demo-home* allows for more controlled data collection, with accurate ground truth location loggings verified by the conductor of the experiment as well as reported by the user. It is also an interesting space in terms of layout, as there is a large corridor where all rooms communicate with, but which does not enable direct transitions between very adjacent rooms as easily as in house setups. At the same time, the *real-home* allows for the collection of data in more naturalistic behaviour,

with the ground truth reported by the user. It is also an interesting case of a house to examine in terms of space usability and semantic location perception: it is a 2-bedroom/2-bathroom flat with an open-plan kitchen-living room space. The bedroom, denoted as `study` and the bathroom, denoted as `loo` constitute an ensuite room. As such, even though there exist 6 different semantic locations (namely `kitchen`, `living room`, `bedroom`, `bathroom`, `study` and `loo`), these can be accounted as 5 or 4 distinct rooms: an open-plan `kitchen/living room`, a `bedroom`, a `bathroom`, and a `study` and `loo`, or a single `ensuite` space. We denoted the above four combinations as *full*, *openplan*, *ensuite* and *merged*, respectively.

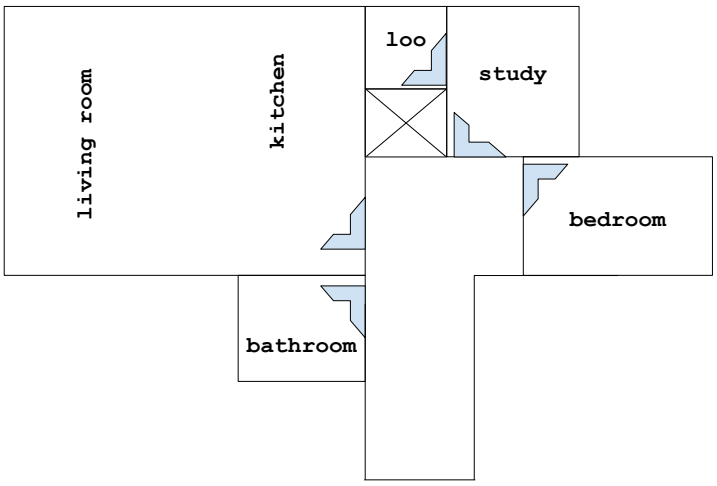
Sensor setup The data were collected by the on-board sensors of a consumer smartwatch, recording ambient BLE signals at 0.2 Hz, and accelerations and angular rates from 6-axis IMUs, and magnetic fields from 3-axis magnetometers at 100 Hz; the smartwatch utilised was a Sony Smartwatch 3 and was attached to the users' dominant wrist. All participants were right-handed.

We examined three possible IoT device configurations: placing beacons at the centre of each room (*centre*), at the entrances of each room (*doors*), and at locations around each room such that their pairwise distances are maximised (*far*). Combinations of the above device configurations might exist in real life, e.g., there might be a smart thermostat by one room's door and a smart speaker in the centre of another; we will only discuss the three configurations above in this thesis, but the dataset includes BLE RSSIs from all the above $3N$ locations simultaneously, allowing researchers to examine other combinations of device configurations should they wish.

Users A total of 11 participants, noted as user1 to user11 performed the experiment across the two environments, yielding a total of 20 recordings, noted as rec1 to rec20 in the dataset. user1 performed the experiment in both environments; three times in the *demo-home* (rec1,5,8) and once in the *real-home* (rec13,15,17,19). user11 performed the experiment only in the real-home (rec14,16,18,20) while the rest of the users performed the experiment only in the demo home, yielding one recording each.



(a) *demo-home.*



(b) *real-home.*

Figure 3.1: Data collection environment layouts.

The floorplans of the two environments where we collected data for our dataset are presented. The *demo-home* consists of $N = 6$ rooms arranged across a single floor. The *real-home* is a 2-bedroom/2-bathroom flat with an open-plan kitchen-living room space. There are 6 distinct semantic locations that, however, can be accounted as $N = 4$, $N = 5$ or $N = 6$ rooms, depending on whether we assume the open-plan kitchen and living room to constitute a single *openplan* room, the ensuite study and loo as a single *ensuite* room, or both.

Note that recording rec13,15,17,19 were recorded simultaneously, but account for the four different assumptions on what constitutes a room in the *real-home*, as described before. The same holds for recordings rec14,16,18,20. Details on each recording are listed in Table 3.1.

Each participant performed the experiment for approximately 1 hour continuously; with the sensor recording application active and the user instructed on how to record the ground truth location, the participants moved around the environment, performing activities that are commonly encountered in each room in their own style. Not everyone performed the exact same activities, and the ground truth activity labels were not recorded.

Table 3.1: Description of **WatchBLoc** dataset.

A total of 11 users performed the experiment across two different environments. `user1` performed the experiment in both the demo-home (for three distinct recording IDs, 1, 4 and 8) and the real-home (recIDs 13, 15, 17, 19).`user11` performed the experiment in the real-home (recIDs 14, 16, 18, 20) and the rest of the users in the demo-home, yielding one recID each. Four versions of a semantic layout for the *real-home* are considered: each semantic location is considered a separate room (*real-home:full*); an ensuite bathroom and bedroom are abstracted to a single ensuite room location in the *real-home:ensuite* case. An open-plan kitchen and living room existing in a space are abstracted to a single semantic room, kitchen/living room in *real-home:openplan*. Finally, both the ensuite and open-plan abstractions are considered in the *real-home:merged* case.

recID	environment	configurations	# rooms
1-12	*demo-home*	*centre*, *doors*, *far*	$N = 6$
13-14	*real-home:full*	*centre*, *doors*, *far*	$N = 6$
15-16	*real-home:ensuite*	*centre*, *doors*, *far*	$N = 5$
17-18	*real-home:openplan*	*centre*, *doors*, *far*	$N = 5$
19-20	*real-home:merged*	*centre*, *doors*, *far*	$N = 4$

3.2.3 Data collection challenges

The collection of this dataset has brought to light a few challenges associated with seamless, privacy-preserving localisation of people. We will discuss some of these here but mention further difficulties encountered in Chapter 4.7, as they appear in the dataset's analysis and algorithm's evaluation.

The BLE RSSI data from different houses may vary significantly, even for homes with similar layouts. Wireless signal strength can be affected by many factors, such as the existence of line-of-sight (LOS) between a BLE device and the smartwatch, attenuation at different levels depending on the construction of the walls' materials [335], etc. As such, it is not always guaranteed that we will hear a BLE RSSI as we would expect in a space; sometimes, RSSI from an IoT device that exists in an adjacent room may be stronger than the RSSI from an IoT device located in the space the user is in. Or, we might have missing data (i.e., not detecting certain beacons at all) even if we're continuously recording data; this can be the result of severe signal attenuation owing to the house's layout (e.g., walls), but also of the user's and other occupants' movements around the house [336], that might be creating additional obstructions to the LOS between device and smartwatch.

A few other challenges arise from choosing the smartwatch as the sole sensing modality. To ensure long battery life, smartwatches may go to battery saving mode, e.g., when their screen is not touched for a long time; this can result in large segments of missing data, leading to uncertain positioning estimates for the corresponding periods of time. At the same time, the wrist attachment of the smartwatch means that the watch is recording inertial data that encode information not only due to the user moving across or within rooms but also when any other activity is performed, as hand usage is entailed in most everyday activities [337]. This leads to much additive "noise" and can significantly complicate distinguishing walking from other vigorous activities, such as brushing teeth or doing the dishes.

Ground truth collection, whilst adhering to privacy concerns, has also been a significant challenge during this experiment's design and data collection. Initial attempts at data collection included ground truth collected through hand-written notes from the user. This was impractical, as the exact timings of changing location and ground truth logs were almost impossible to align, and it was heavily relied on the participants to remember their actual course of action. Camera use was prohibited, as data collection included performing the experiment at participants' private houses and was thus considered a privacy breach by the Ethics Approval

Committee. RF tags were also inappropriate to indicate the ground truth location of the participant, as multiple people were using the data collection environments simultaneously. Based on these issues, a logging application was developed and installed on the smartwatches so the participants could tap the appropriate semantic location on the smartwatch's screen every time they moved to a new room. It is important to note that this approach can still have synchronisation issues, as it is almost impossible for the participant to tap their new location exactly at the moment of room-change. However, exploration of the data has shown that it is reasonable to assume that the ground truth lags are no longer than approximately 5 seconds.

In Chapter 1.4, we demonstrate the capabilities of this dataset to perform semantic localisation in indoor spaces, accurately capturing room transitions across both environments.

3.3 The watchHAR dataset

In this section, we will present watchHAR, a new dataset of smartwatch IMU recordings of users performing ADL commonly performed in indoor spaces, e.g., a home or office environment, coupled with location information from a VICON system. The dataset is released as part of this work at the following address:

`https://zenodo.org/record/7092552`

3.3.1 Motivation

Human Activity Recognition has been an active research area for many years, with datasets available for various sensor modalities, including cameras, IMUs, smartphones, smartwatches, and other body-worn sensors. However, there remains a limited presence of datasets that focus specifically on smartwatch recordings, despite their advantages in ensuring continuous movement tracking, being lightweight, and widely available. Additionally, many HAR datasets emphasise health-related events, such as falls and gait patterns, or capture low-level actions, such as reaching for

objects or opening and closing drawers, rather than broader activities of daily living (ADLs).

Most existing datasets primarily rely on smartphone data and often cover a restricted set of activities. For example, the UCI HAR dataset contains only six locomotive activities recorded via smartphone sensors. The Opportunity dataset, while rich in sensor data—including smartphone and smartwatch inertial recordings, focuses on fine-grained actions rather than ADLs. The ExtraSensory dataset provides in-the-wild recordings from both smartphones and smartwatches but suffers from missing labels, annotation inconsistencies, and unbalanced class distributions. Among the few datasets that offer accurate smartwatch recordings and capture a diverse range of ADLs are PAMAP2 and WISDM.

Our dataset contributes to this space by specifically addressing the need for smartwatch-based HAR recordings focused on ADLs, particularly in private indoor environments such as homes and offices. This is especially relevant given the rise in remote working since the COVID-19 pandemic, where understanding everyday activities in these settings is increasingly valuable.

3.3.2 Dataset Description

Sensor setup The data were collected by the on-board sensors of 4 Sony Smartwatch 3 wearables, one attached per wrist and ankle of the users, recording accelerations and angular rates from 6-axis IMUs, and magnetic fields from 3-axis magnetometers.

A Vicon motion capture system was deployed to produce high-precise groundtruth values of the user as they were performing each activity, providing not just semantic description of each activity (based on its given annotation), but also precise movement trajectories captured in the vicon system.

A VICON marker was attached to the users' wrists, just below the smartwatches, to provide location information of the users' hand movements. We also attached markers on the participants' ankles to provide synchronisation information about the user's hand-swinging motion and the steps taken. Vicon markers are small,

retro-reflective markers attached to a subject's body or object that is tracked by a Vicon motion capture system, allowing for precise 3D tracking of movement, enabling the system to pinpoint the location of a specific point on a moving object.

A GoPro camera was utilised to record the users executing the prescribed activities, to ensure any deviation from the instructions is noted and to ensure accurate ground truth labels.

Selected activities 18 activities were recorded, selected from four key categories of activities frequently performed day-to-day: walking patterns, floor changes, relaxing, and activities involving manual dexterity. These are described in Table 3.2, grouped by their movement style similarities, where movement patterns aiming to change location (walk, change floor) are disambiguated by static activities; these, in turn, are disambiguated into activities involving little to no hand involvement, to active manual dexterity. We believe this provides a variable and diverse range of activities that commonly occur in daily living settings, and thus are useful to consider.

Users Both male and female users, aged between 23 and 67 years old, participated in the experiment. The users were instructed on the activity to perform but were told to execute it in their own style.

We performed two experiments: in the first experiment, 13 users executed 14 of the 16 activities described in Table 3.2 in their own style in a room equipped with VICON cameras. The two activities not performed in the above experiment were "walking upstairs" and "walking downstairs" because stairs were unavailable in laboratory spaces equipped with VICON cameras. As such, the two "stairs" activities are the only activities in the dataset for which location information from the VICON system is not provided. In the second experiment, 14 users performed the remaining two activities on stairs of varying lengths and floor numbers, recorded in the smartwatch's IMU recordings. Six out of the 14 users had also participated in the WatchBLoc experiment. The users performed each activity for approximately 1 to 3 minutes.

Table 3.2: Description of **watchHAR** dataset

group	label	description
walk	freewalk	normal walk (free-swinging hands)
	bag	walk holding a bag
	phone	walk while talking on phone
	pockets	walk with hands in pockets
	tray	walk holding a tray
	underarm	walk holding an object under armpit
change floor	up	walk upstairs
	down	walk downstairs
relax	relax	sit read a book use a remote control
do manual tasks	type	type on a keyboard
	write	write on a paper
	hands-face	wash hands and face
	teeth	brush teeth
	mug	wash a mug
	plate	wash a plate
	sandwich	prepare a sandwich

While smaller in size than other similar datasets, e.g., PAMAP2, WISDM, the dataset contributes a useful range of activities of daily living, that, as we will explore in Chapter 5.7 and Chapter 6.7, can be utilised to learn robust models of activity recognition systems. At the same time, it includes precise trajectory-based information from the Vicon system, paving the way for usability also in other domains, such as pose estimation and trajectory-based HAR.

3.4 Conclusions

In this chapter, we introduced **WatchBLoc** and **watchHAR**, the two datasets we have collected to assist the work in this thesis and the research community, along with challenges identified during the data collection process. In the following chapters, we will use these datasets to develop, train and evaluate our localisation and activity recognition algorithms.

Chapter 4

Room-level localisation in privacy-preserving environments

4.1 Introduction

In this chapter, we will explore the first of the three key components that contribute to human behavioural understanding, estimating a person's location. We will investigate the problem of human positioning in indoor spaces, where preserving the privacy of the user is key, and available infrastructure to allow for effective localisation is limited.

4.1.1 Motivation

In recent years, people spend a considerable amount of their day in their homes. An increased tendency towards self-employment, as well as businesses experimenting with new productivity schemes, have given more flexibility to people to work from home, a trend that has largely culminated in a necessity due to the COVID-19 pandemic. At the same time, the advances in medicine and the increase in life expectancy have resulted in the phenomenon of the ageing population; retired adults typically have many more years to live but are often faced with age-related diseases that might keep them at home as they become more severe.

The increasing at-home way of living presents a unique opportunity to analyse human behaviour at home to improve situation awareness for Cyber-Physical Systems (CPS) that function based on human-environment interactions. For example, smart homes could benefit from daily patterns to learn tasks, such as preheating a room frequented at a usual time. Similarly, older adults could be remotely supervised, and intervention could be called for should an abnormal incident be detected.

By investigating such scenarios, it becomes evident that human-environment interactions at home correlate with semantic locations around the house and the user's mobility patterns around different rooms. As such, inferring the position of a human inside their house, as well as transition patterns across rooms, can significantly improve situation awareness.

4.1.2 Challenges of human positioning in private indoor spaces

Human positioning at home is far from trivial. Installation of bespoke infrastructure can be cumbersome and cost-prohibitive. Privacy concerns can also be raised, not only regarding the person monitored but also for others who might be sharing the space. It is, therefore, essential to identify methods that can exploit existing infrastructure to infer space usage and effectively capture mobility patterns in a privacy-preserving way. Despite the surge of research activity in recent years in the area of human indoor positioning, these constraints rule out many methods that are very effective in different contexts.

User privacy, in particular, which is a key motivation for this work, plays a critical role in determining the selection of sensor types and methodologies for indoor positioning. Privacy is a multifaceted concept encompassing various directions, from the non-disclosure of personal attributes (e.g., age and gender) to biometric data such as heart rate, facial structure, voice, and gait patterns, as well as information related to financial or social status and online data security [338, 339]. In this thesis, user privacy is defined in terms of three primary aspects: i)

personal attributes, ii) biometric data, and iii) information revealing financial status, occupation, or sensitive category data, such as racial or ethnic origin, political or religious beliefs. While concerns regarding data security are particularly relevant in this context [340–342], where ambient and wearable data are collected, stored, and processed either on-device or on designated remote servers, as we will discuss in detail in the following sections, this work does not address data privacy from the viewpoint of cyber security and privacy in computing systems. Instead, the focus is on the algorithmic and infrastructural components required to achieve seamless user localisation without exposing identifiable or personal characteristics.

To this end, camera-based approaches, despite their proven accuracy in human positioning [25, 90, 91, 95–98, 101–108], are unsuitable for privacy-preserving localisation. Such methods can capture images or videos of individuals, revealing biometric features (e.g., facial characteristics and gender) or details of their surroundings (that may disclose financial status or personal affiliations). Similarly, sounds-based localisation systems, that record voice (which belongs to biometric data) cannot be utilised [113–118]. At the same time, positioning technologies requiring access to the user space for infrastructure installation, such as infrared systems, or detailed housing information, such as architectural maps needed for many radio-frequency-based fingerprinting approaches, are also not appropriate. While IMU-based methods align with most privacy constraints, gait patterns remain an identifiable biometric characteristic, and mobility algorithms should therefore avoid reliance on gait estimation [138, 152, 154–158, 160], such as approaches employing pedestrian dead reckoning.

Given these considerations, the remainder of this section explores key positioning methodologies from prior research, evaluating their constraints in terms of user privacy, cost, and suitability for human localisation in private indoor spaces such as homes and offices.

Infrared (IF) positioning systems, for instance, though capable of providing very accurate position estimates, require the installation of extensive infrastructure, making them cost-prohibitive for the home environment case [121]. Pedestrian

Dead Reckoning (PDR) is an alternative that could minimise costs significantly, as it only requires cheap inertial sensors (accelerometer, gyroscope, magnetometer) that can also easily be attached to the person to be monitored and are widely available in everyday devices, such as smartphones and smartwatches; however, it requires very accurate knowledge of one's initial position, and it can also exhibit significant errors due to drift [337]. This issue is exacerbated in smartwatches where there could be interference from multiple other non-walk-related hand motions [337]. Other radio frequency (RF) localisation techniques have successfully been used in indoor public spaces (e.g., shopping malls) that are equipped with large wireless sensor networks (WSN/WLAN) with numerous access points (APs). These methods usually employ Wi-Fi APs or Bluetooth low energy (BLE) beacons [29] together with smartphone inertial measurement unit (IMU) sensors [30] to achieve an accuracy with an estimated $2 - 10$ metres error [30]; errors beyond 4 m however would probably correspond to a completely different room in a house. Furthermore, they usually require extensive fingerprinting or crowdsourcing [175], which is not an option at home. Lastly, even though mobile phones are carried very often by their owners wherever they go, it is not common for people to carry the device at all times inside their houses. Older people, in particular, are often unfamiliar with and resilient to use such technology. Device-free passive RF localisation methods that do not require any sensors to be carried could probably be effectively used to monitor people who live alone. Still, they will raise difficulties in more crowded houses, as there is no way to distinguish between the movements of the house's occupants (or visitors) if they are not carrying a tag and require careful training per house.

4.1.3 Proposed approach and key contributions

Smartwatches are equipped with inertial sensors and Bluetooth radio, with which they can sense IoT devices in their vicinity. As these are seeing increasing use among the wider population, we assume this to be a practically infrastructure-free approach.

In summary, our main contributions are:

- We propose **RoomE**, a probabilistic algorithm that achieves semantic localisation and room identification by sequentially fusing ambient location information with high-level mobility features derived from inertial data, no matter the IoT devices' placement in the environment.
- We investigate the challenges associated with the sensors we use in this study: i) how different placements of IoT devices around the house can affect the accuracy of room detection, and ii) the challenges associated with using motion data from smartwatches and the need for a simple yet robust approach to detecting candidate room change events.
- We evaluate our proposed approach on the WatchBLoc dataset across different floorplan layouts, IoT device placements and user motion patterns. We show that our method mitigates the high sensitivity of existing smartwatch room detection algorithms to the configuration of ambient home devices. It increases the room detection accuracy compared to state-of-the-art from 67.77% to 81.53% without using fingerprinting of the environment. It also provides the capability of inferring room transition probabilities without using floorplan information.

The remainder of this chapter is organised as follows: Section 4.2 presents the architecture of our proposed indoor localisation system and illustrates how the different modules work together. Section 4.3.3.2 introduces the algorithmic details for each of the system's components. Section 4.4 then prepares the data to be used by the framework, and Section 1 evaluates their performance. Finally, Section 4.7 concludes the chapter and discusses limitations and future directions.

4.2 System architecture

This section provides a high-level description of our system architecture and its three main algorithmic components, i.e., **Room Detection**, **Walking Detection** and **Room Estimation**, each of which will be detailed and evaluated in the following sections.

Our algorithm assumes a house with N rooms, each equipped with exactly one IoT device. In practice, this is not constraining in the presence of multiple IoT devices per room, as they can be aggregated into an, e.g., 'max' or 'mean' BLE signal. Handling rooms with no BLE devices is more challenging and is an open topic for future work. We also assume that we can infer a semantic label for the room each IoT device is located in through its identifiable name, e.g., an "IoT fridge" is located in the "kitchen", etc. The inhabitant owns a smartwatch, which can perceive these devices through their emitted signals. No information is available regarding the size of each room, their relative position in the house or the position of the IoT devices within the rooms.

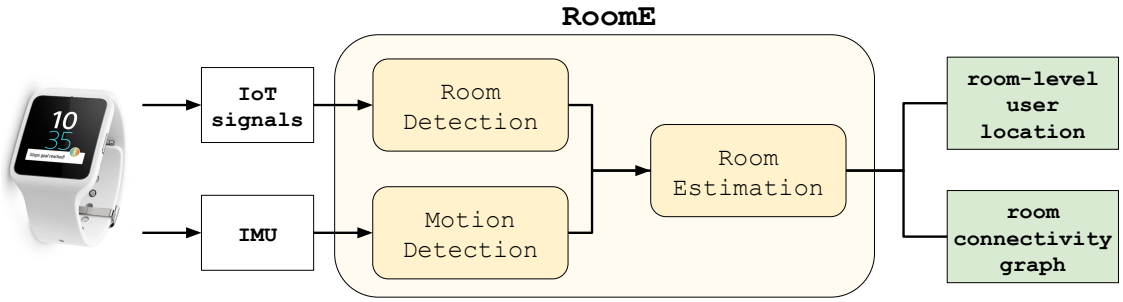


Figure 4.1: System architecture for **RoomE**.

Our algorithm aims to estimate the room-level location of the inhabitant and a semantic map of the environment while they freely move around the house. This is achieved through **RoomE**'s three main components. These are depicted in Fig. 4.1 and outlined below:

1. **Room Detection**, giving initial estimates of the user's semantic location.
2. **Motion Detection**, to estimate potential room-transitions.
3. **Room Estimation**, to provide a definite, improved room-level user location and information on the house's layout.

The first two modules utilise the ambient IoT signals from the various IoT devices and the inertial data logged in the smartwatch to derive initial estimates of the user's room-level semantic location and mobility, respectively.

In the final module, we employ a probabilistic graphical model, which we train against the available data, to fuse the information extracted in the above modules in a probabilistic graph. Traversal of the graph in the inference phase provides refined room-level localisation for the user. The learnt model's parameters during the training phase are used to infer room-connectivity information, i.e., which rooms are accessible from others.

RoomE thus outputs a room-level location for the user at each timestamp, as well as an estimated semantic map of the house, in the form of a room-connectivity graph.

4.3 Algorithms

This section discusses the three main algorithmic components of **RoomE**.

4.3.1 Room Detection

For the **Room Detection** module, we employ a *maxRSSI* approach on the preprocessed BLE data, thus assuming that at any point, the user is nearest to the IoT device that emits the strongest RSSI and thus in the corresponding room in the house where the IoT device is placed. BLE data preprocessing is discussed in Section 4.4.1.

Other algorithms that can infer semantic location from ambient IoT data could be used instead of *maxRSSI*. Our choice of *maxRSSI* is based on the constraints that the home environment poses: we do not have any information about the dimensions of the rooms or the IoT devices' placements around the house; this makes the use of, e.g., fingerprinting based algorithms less suitable for our problem. Even if we did have such information, we would need to re-calibrate the model in each house, as the signal propagation would be affected by the specific conditions of each environment.

maxRSSI is a simple approach that does not require prior training or calibrating, as it only concerns the strongest RSSI signal each time. It has been considered accurate enough to generate the ground-truth location of users in human-robot interactions [171], as well as to identify the interactions of users with objects

that have BLE beacons attached [343]. It also resembles the thought process behind log-linear path-loss radio propagation models, which are often employed to model the degrading effect of the distance between the transmitter and the receiver on the RSSI values:

$$rssi_d = rssi_{d_0} - 10n \log_{10}(d),$$

where $rssi_d$ is the RSSI at a distance d, $rssi_{d_0}$ is the received signal power of the receiver from a transmitter one meter away, and n is the path loss exponent. As is obvious from the above, for $d_1 > d_2$, it holds that $rssi_{d_1} < rssi_{d_2}$, i.e., the RSSI value decreases as the distance from the transmitter increases. As such, the maximum RSSI value will be obtained nearest the transmitter.

The data passed to the *maxRSSI* algorithm need to be vectors of RSSIs, corresponding to the N RSSIs received from the N identified rooms in the house at each timestamp t (i.e., the end-time of the corresponding window):

$$vecRSSIs_t := [rssi_{r_1,t}, \ldots, rssi_{r_N,t}] .$$

The room estimate r_t can then be computed as the room where the strongest signal device is located. For windows that, even after the preprocessing, no signal was received, we cannot accurately estimate the user's location with the *maxRSSI*. As such, we assign these windows to an "unknown room":

$$r_t = \begin{cases} \text{unknown room}, & \text{if no RSSI from beacons,} \\ r_j,\ j = \operatorname{argmax} vecRSSIs_t, & \text{otherwise.} \end{cases}$$

4.3.2 Motion Detection

For the **Motion Detection** module, we employ a simple *energyPeak* approach based on the assumption that the overall energy of the acceleration signal recorded in the smartwatch should be higher during active phases compared to idle phases.

Many fitness-related smartwatch apps aim to provide various information about mobility, including step counters or activity recognition apps; however, these can be unreliable for the home environment case-study [344]. Smartwatches can effectively

identify vigorous activities, such as switching between walking, cycling and running, but not so well activities performed at lower speeds [345] or combined with other domestic activities [344]. At the same time, smartwatch step counters commonly log many false-positive steps [346]; this might not be problematic for distances typically travelled outdoors, but it can be limiting when trying to estimate mobility within a house.

Regardless, for our scenario where floorplans of the house are also absent, a step-counter would have little benefit: as we are unaware of the size of the various rooms, as well as of their proximity, using a step counter as the mobility estimate would be inappropriate, as it would require assumptions about the number of steps that connect different rooms. It would also make it more difficult for the model to generalise across different houses.

Thus, we suggest that to have a system that easily applies to every environment and any smartwatch and, therefore, requires no calibration from the user, a much coarser mobility estimate would better suit our needs. High-level mobility estimates, such as walking event detection, are well-suited for this scenario. Numerous activity recognition models achieve high accuracy in detecting walking activities, as is detailed in Chapter 5.7, and can be adopted if the aforementioned issues of lower-speed-execution or multitasking of activities that arise in the domestic scenario [344, 345] are not hindering their performance. This work adopts a proof-of-concept approach, employing simple and lightweight mobility estimates. We propose a threshold-based method for motion detection inspired by public health research on physical activity identification. While not necessarily the most sophisticated option, it effectively demonstrates the core principles of our method, and more complex models could be integrated if required.

For our *energyPeak* approach, we first further filter the acceleration signal to the human walking frequencies, i.e., $[1.2, 2]$ Hz, to remove any redundant frequencies related to other activities. We then calculate the energy of the filtered signal through the Short-Time Fourier Transform (STFT) of the acceleration's magnitude, calculated in windows of $512 = 2^9$ samples, i.e., 5.12 sec. The choice of a 5.12 sec is

two-fold; first, BLE RSSI data in our dataset (and often in wearables) are recorded at 0.2 Hz, meaning that we have approximately a sample in each 5 sec window. At the same time, STFT favours window sizes that are a power of 2; as a result, we select 5.12 sec as the first window size that satisfies both requirements. Prior work in using 5.12 sec for HAR also supports this choice [347]. This ensures a synchronised 1-to-1 mapping between location and mobility estimates. As we observed that high-energy contents gradually occur around ground truth transitions, we chose the envelope of the calculated energy as a final smoothed-out energy estimate.

The motion estimates at time t are then chosen as the local peaks (calculated with the `MATLAB`'s `findpeaks` function) of the normalised energy that are larger than the energy's average:

$$m_t = \begin{cases} \text{active}, & \text{if } acc_{\text{energy,smooth}_t} \text{ is a peak and} \\ & \quad acc_{\text{energy,smooth}_t} > \text{threshold}, \\ \text{idle}, & \text{otherwise.} \end{cases}$$

Peak detection using a predefined amplitude threshold is a widely used method for step and walking event detection [348–353]. Thresholds are often determined through exhaustive search within the signal's amplitude range in the time domain, while some approaches, including ours, operate in the frequency domain. However, most methods are specific per device utilised [350], require prior access to walking data and their true/false positive rates for optimisation of the threshold selection, or employ a heuristic approach, exhaustively searching within the range of minimum and maximum variance of the data [354, 355]. This work adopts the approach proposed in [353], where walking event detection is based on signal energy, with the threshold defined as a scaled version of the mean signal energy. Although [353] employs a different signal energy calculation (the Teager-Kaiser Energy), this approach remains a simple and adaptable threshold selection mechanism that does not require extensive data preprocessing, facilitating deployment in diverse environments and users.

4.3.3 Room Estimation

The **Room Detection** and **Motion Detection** modules have exploited the ambient IoT signals sensed from the smartwatch and its IMU to provide initial location

estimates and estimates of walking events. In this step, we fuse this information to improve the **Room Detection** module.

To fuse these two types of observations (r, m) - i.e., ambient location and mobility estimates - and thus derive refined room-level estimates, we take advantage of the inference capability of an appropriately defined Hidden Markov Model (HMM) with parameters $\theta \equiv (E_R, E_M, T, \pi)$, corresponding to the room emission probabilities E_R, the motion emission probabilities E_M, the state transition probabilities T, and the initial state vector probabilities π. We then refine our initial estimate by the parameters of the model that match our data for each house.

4.3.3.1 Model definition

We define the state space of our HMM using "one hot encoding" for the set of all possible N^2 room transitions. We choose this room-transitions-as-a-state representation, as our mobility estimate m is concerned with likely room transitions. As such, a state at timestamp t is defined as:

$$s_{k,t} := (r_{i,t-1} \rightarrow r_{j,t}), \text{where } i, j \in \{1, 2, \dots N\},$$
$$k := j + (i-1)N,$$

where k is indexing the hidden states of the markov model. In this thesis, we index all states representing room transitions from room $r_{i,t-1}$ to any other room $r_{j,t},\ j \in \{1, 2, \dots, N\}$ sequentially in an increasing order. For example, consider we are at $r_{1,t-1}$; then, $s_{1,t} = (r_{1,t-1} \rightarrow r_{1,t})$, $s_{2,t} = (r_{1,t-1} \rightarrow r_{2,t})$, ..., $s_{N,t} = (r_{1,t-1} \rightarrow r_{N,t})$. Then, the following N indices will correspond to transitions from $r_{2,t-1}$ to any other state, $s_{N+1,t} = (r_{2,t-1} \rightarrow r_{1,t})$, $s_{N+2,t} = (r_{2,t-1} \rightarrow r_{2,t})$, ..., $s_{2N,t} = (r_{2,t-1} \rightarrow r_{N,t})$, and this is repeated until the final state, indexed as to $s_{N^2,t} = (r_{N,t-1} \rightarrow r_{N,t})$.

We also define the observation space using the sensor measurements obtained from the above modules, i.e., as the tuple $o_t := (r_t, m_t)$ at a given timestep. We assume that r and m are independent from each other.

To refer to a room-transition pair and a combination of observations pair irrespective of time, we will use the following notation:

$$s_k \models (r_i \to r_j), \text{where } i, j \in \{1, 2, \ldots, N\},$$
$$k := j + (i-1)N,$$

and

$$o \models (r, m), \text{where } r \in \{r_1, r_2, \ldots, r_N\},$$
$$m \in \{idle, active\},$$

assuming that a state is always defined across two consecutive room locations in time, and a pair of observations is only properly defined when these coincide.

The emission probabilities for each type of observation are detailed in the following paragraphs.

Room emissions We initialise the room emissions E_R based on the assumption that when we move to a new room, we expect the strongest RSSI to be emitted from the IoT device located in the room we arrive in. This includes the case when we stay in the same room in two consecutive time windows, as this event is encoded as a transition from a room to itself in the states of the HMM. Note that we have N rooms, but we can also observe the 'unknown room'; we can thus have $N + 1$ room observations.

The probability $p(r|s_k)$, i.e., the probability of observing room r when moving from r_i to r_j, is thus the following:

$$E_{r,k} = \begin{cases} 1 - \varepsilon_r, & \text{if } r_j \equiv r, \\ \varepsilon_r \ / \ N, & \text{otherwise.} \end{cases}$$

The above values are chosen to reflect our initial assumption that when moving into a new room, the beacon existing in this room will *almost surely* emit the strongest RSSI. The value ε_r is chosen to represent the uncertainty of the strongest RSSI being emitted from within the room of occupancy. We initialise $\varepsilon_r = 0.01$ for our experiment to reflect the expected RSSI behaviour and for numerical stability.

Motion emissions The motion emission matrix must reflect that people are likely to be moving within a room without changing their semantic location, but not the opposite: a user cannot change their location without moving. As such, we define the probability of observing motion m when moving from r_i to r_j, i.e., $p(m|s_k)$, as:

$$E_{m,k} = \begin{cases} \begin{cases} \varepsilon_{m,c}, & \text{if } m : idle \\ 1 - \varepsilon_{m,c}, & \text{if } m : active. \end{cases}, & \text{if } s_k = (r_i \to r_j),\ i \neq j, \\ \begin{cases} 1 - \varepsilon_{m,s}, & \text{if } m : idle \\ \varepsilon_{m,s}, & \text{if } m : active. \end{cases}, & \text{otherwise.} \end{cases}$$

The values $\varepsilon_{m,c}, \varepsilon_{m,s}$ are chosen to model our prior on the user's mobility. Depending on the application and our knowledge of the environment, these can be selected to be identical in all rooms or reflect the different mobility patterns in each room (e.g., we might expect higher mobility in a kitchen compared to a study). In this work, we assign $\varepsilon_{m,c} = 0.01$ to reflect the certainty of movement during room-changes. In reality, this probability equals 0; there is no way to move from one room to another without a walking event. However, we chose to allow a very low probability of $E_{idle,i\neq j} = 0.01$ to avoid numerical instabilities in the code. To initialise the states representing a "stay-in-the-same-room" transition, we assign $\varepsilon_{m,s} = 0.3$, as it is still possible to walk within a room without any intention of changing location. Research has shown that people are expected to be sitted at least 75% of their time when working [356], so we define the above probability loosely based on these statistics.

Initial state vector The initial state probabilities vector π is defined by assuming we start uniformly at random from any state representing a stay-in-same-room transition:

$$\pi_k = \begin{cases} 1 \ / \ \mathrm{N}, & \text{if } s_k = (r_i \to r_i),\ i \in \{1, 2, \ldots, N\}, \\ 0, & \text{otherwise.} \end{cases}$$

Transition matrix To define the state transition matrix T of the HMM, we first need to remember that each state represents a *room transition* from room r_i to room r_j, i.e., $s_k \models (r_i \to r_j)$. The transition matrix T of the HMM is thus defined subject to the constraint that the first part of the following state must agree with the last

part of the current state. Any state-transition that adheres to this constraint has the same probability. Any other state-transition has zero probability:

$$T_{fg} = \begin{cases} 1 \ / \ \mathrm{N}, & \text{if } s_f = (r_i \rightarrow r_j),\ s_g = (r_q \rightarrow r_p) \text{ and } j \equiv q, \\ 0, & \text{otherwise.} \end{cases},$$

where $f, g \in \{1, 2, \ldots, N^2\}$ and $i, j, q, p \in \{1, 2, \ldots, N\}$.

Note that should we have priors about room occupancy, e.g., known behavioural patterns of an individual, or access to room occupancy statistics from relevant studies, the initial state and room transition probabilities can be adapted to reflect these and accelerate convergence. In lack of such statistics in this work, we default to uniform probability distributions.

4.3.3.2 Parameter learning

This step aims to improve the model's parameters by learning the optimal values that best describe the observed data. Once we have the learnt model parameters, we can use the updated model to infer a refined estimate of the user's room-level position.

One key detail in our approach is that we do not need to learn the full set of the model's parameters. This can be justified by recalling that we want to use the mobility data m to *correct* the **Room Detection** estimates; we thus need the model to expect a motion event when in states that represent room-transitions but not room-stays. As such, we can keep our assumption regarding whether an idle or active event should ignite a room transition fixed over time. We also need our model to work its way through 'unknown room' occurrences by only trusting the *energyPeaks* estimates in such a case; thus, the 'unknown room' observation should maintain its low probability occurrence at all times.

We thus train the room emissions $E_{r,new},\ r \neq$ 'unknown' and the state transitions T_{new} only, with the Baum-Welch algorithm for expectation-maximisation (EM). Updating the state transitions might also be optional, as the transition matrix has been designed to indicate only allowed or not allowed transitions. However, learning this transition matrix can provide information about room connectivity.

The standard Baum-Welch algorithm (see [357] for the full set of equations) is adapted as follows:

For the E-step:

- In the forward pass, the observed emissions can be modelled as the product of the corresponding elements of the two emission matrices, E_R, E_M, as these observations are independent:

$$\alpha_{t,k} = \sum_{f=1}^{N^2} \alpha_{t-1,f} T_{fk} E_{r_t,k} E_{m_t,k},\ k \in \{1, 2, \ldots, N^2\},$$
$$r \in \{r_1, r_2, \ldots, r_N, \text{“unknown room”}\},\ m \in \{idle,\ active\}.$$

  where each forward probability $\alpha_{t,k}$ represents the joint probability of being in state $s_{k,t}$ at time t and observing the first t observations by time t, knowing the model parameters θ. It holds that $\alpha_{0,k} = \pi_k E_{r,k}(r_0) E_{m,k}(m_0),\ k \in \{1, 2, \ldots, N^2\}$.

- Similarly, the backward pass is defined as follows:

$$\beta_{t,f} = \sum_{f=1}^{N^2} T_{fk} \beta_{t+1,k} E_{r_{t+1},k} E_{m_{t+1},k},\ k \in \{1, 2, \ldots N^2\},$$
$$r \in \{r_1, r_2, \ldots, r_N, \text{“unknown room”}\},\ m \in \{idle,\ active\}.$$

  with $\beta_{L-1,k} = 1,\ k \in \{1, 2, \ldots N^2\},\ L$: the total number of timestamps.

- The probabilities $\gamma_{t,k}$ of being in state $s_{k,t}$ at time t and $\xi_{t,f,k}$ of being in state $s_{f,t}$ at time t and state $s_{k,t+1}$ at time $t+1$, are defined as standard, subject to the above modifications of $\alpha_{t,k}, \beta_{t,f}$.

For the M-step:

- As discussed, we only need to learn the room emission probabilities. As such, each element $E_{r,k_{new}}$ in the updated emission matrix is the normalised expected number of times in state s_k and observing r, i.e.:

$$E_{r,k_{new}} = \frac{\text{expected \# times in } s_k \text{ and observing } r}{\text{expected \# times in } s_k},\ r \in \{r_1, r_2, \ldots, r_N\},$$
$$k \in \{1, 2, \ldots, N^2\}.$$

Note that the value for $r_i \equiv$ "unknown room" remains unchanged. We ensure that the full E_R matrix is always properly normalised to ensure the numerical stability and feasibility of the algorithm. The motion emission probabilities E_M also remain unchanged.

- To update the state-transitions, each element $T_{fg_{new}}$ in the updated transition matrix is the normalised expected number of transitions from s_f to s_g, i.e.:

$$T_{fg_{new}} = \frac{\text{expected \# transitions from } s_f \text{ to } s_g}{\text{expected \# transitions from } s_f}, \; f, g \in \{1, 2, \ldots, N^2\}.$$

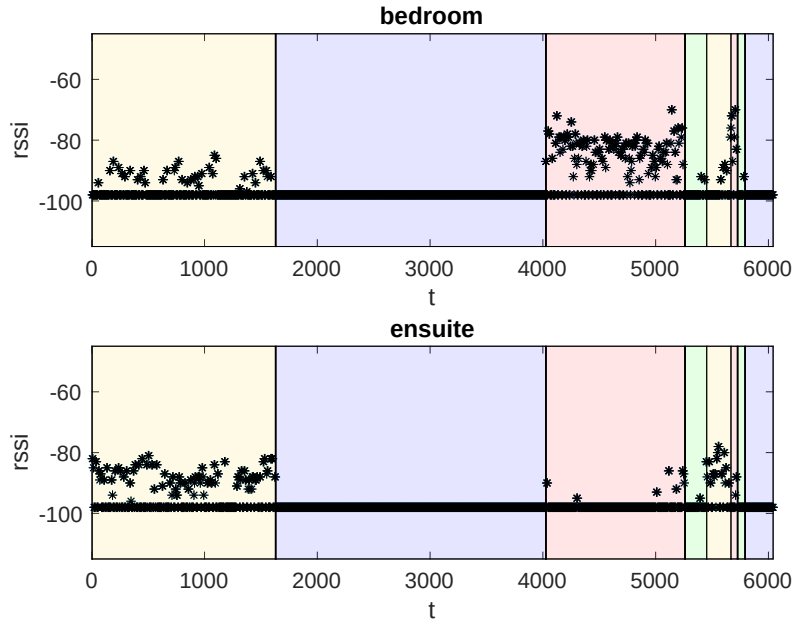
4.4 Data preparation

In this section, we describe the necessary preprocessing of the WatchBLoc data to be used by RoomE.

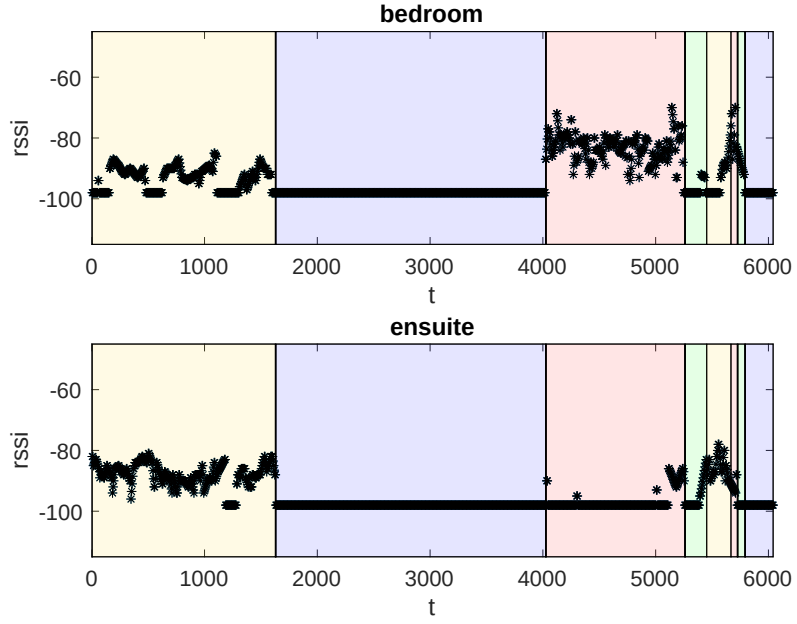
4.4.1 BLE data

A basic preprocessing was used to prepare the BLE RSSI data from the WatchBLoc dataset for the **Room Detection** module.

First, we segment each BLE RSSI signal in non-overlapping windows of length 5.12 sec. The choice of the window size is related to the 0.2 Hz recording frequency of the BLE data, as well as an implementation detail in our approach for the **Motion Detection** module that eases time synchronisation of the two modalities. When multiple RSSI logs from the same device exist in a 5.12 sec window, these are aggregated to a single RSSI value, the maximum among them. The timestamp t for the data in each window is thus now updated to be the end-time of the window. The BLE data are linearly interpolated to deal with missing "no-signal" RSSI values, following the 5.12 sec max-aggregation. Fig. 4.2 shows an example of the behaviour of RSSIs from two different rooms in *real-home*, as well as the effect of the preprocessing on the final BLE data, which are used by the rest of the **Room Detection** module.



(a) Raw BLE signals.



(b) Preprocessed BLE signals.

Figure 4.2: Example of raw and preprocessed BLE RSSIs across rooms.

BLE RSSIs logged in two adjacent rooms, `bedroom` and `ensuite`, in the *real-home* environment of our dataset. The coloured areas denote the ground truth room the user was in at each time, each colour corresponding to a separate room; pink and yellow, in particular, correspond to the ground truth bedroom and ensuite locations, respectively. We observe that we can register RSSIs from multiple BLEs in adjacent rooms. Also, raw BLE RSSI recordings (top) can have missing data. Interpolation can help in smoothing intermittent no-signal phases (bottom).

4.4.2 IMU data

A basic preprocessing was used to denoise, normalise and prepare the **WatchBloc** data for the **Motion Detection** module.

Similarly to the BLE data, we first perform linear interpolation to fill any missing values. Then, we perform moving median filtering to achieve an initial reduction of noise, which is mainly present due to the sensor being on the user's wrist. A high-pass Butterworth filter with cut-off frequency 0.3 Hz is then employed to remove the gravity vector from the acceleration data, and a low-pass Butterworth filter with cut-off frequency 20 Hz (that corresponds to the range of frequencies of human activities) is used to further denoise the data [358]. The data are then bounded to $[-1, 1]$ and normalised to zero mean and unit standard deviation. The processed data can now be used from the **Motion Detection** module, as described in Section 4.3.2.

4.5 Evaluation

This section presents the experimental evaluation of our system's components.

4.5.1 Room Detection: the *maxRSSI* baseline

We evaluate the *maxRSSI* approach that we defined in Section 4.3.1, for the **Room Detection** module on the BLE RSSI data and the corresponding ground truth semantic location, as they were collected in our dataset.

Fig. 4.3 demonstrates the performance of *maxRSSI* in room-level localisation for the two environments where the **WatchBLoc** dataset was collected, shown in Fig. 3.1, and for the three beacon setups in the space, namely *centre*, *doors* and *far*.

A few interesting observations are in order: first, we verify that the device configuration within the space indeed matters. The performance of *maxRSSI* is largely variable between the various configurations for most users. For some users, performance might have less variability between configurations; we believe this is related to which areas of the room each user covered when performing the experiment (recall that there was no prescribed path the users should follow). Second, it seems

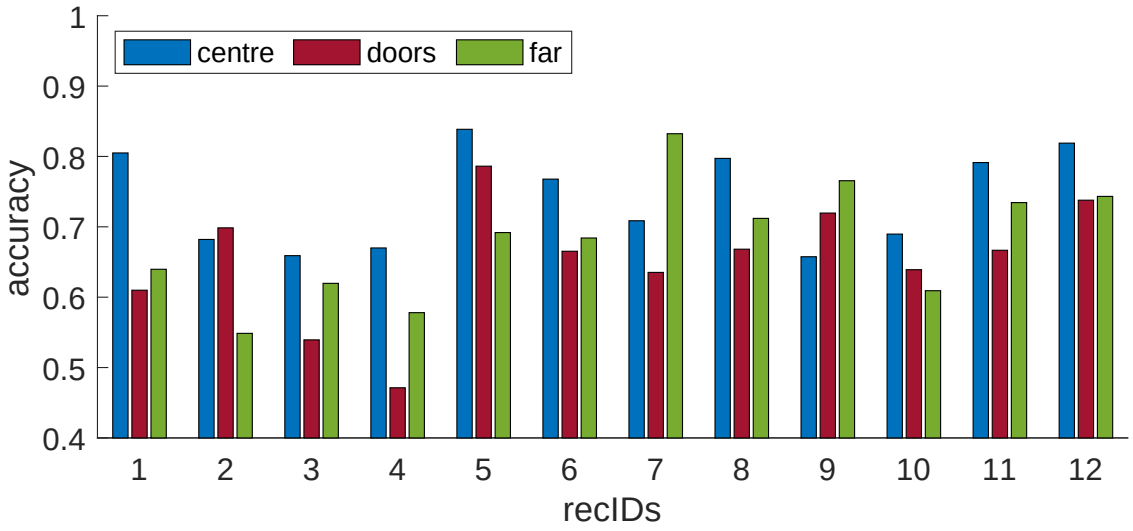
there might not be a global optimal configuration placement for all houses; we are only considering two environments in this work, and thus we cannot be conclusive, but the *centre* device configuration is dominant in the *demo-home*, in contrast with the *far* configuration which is the best performing in the *real-home*. Overall, the *maxRSSI* average accuracy is 67.77%.

It is interesting to investigate whether we can attribute the low performance of *maxRSSI* to certain conditions. Fig. 4.5a demonstrates the room-level locations estimated with *maxRSSI* for a case in the dataset where the *maxRSSI* approach is significantly low-performing: recID 19 − *centre* with just 57.03% accuracy. As is apparent, the recording includes long segments where no BLE RSSIs have been logged. It is also a case where two devices placed in separate rooms are sensed simultaneously, with the strongest RSSI value often heard from the adjacent room and not the one the user is in. We revisit this recording in Section 4.5.2 to demonstrate the improvements achieved in such a challenging case.

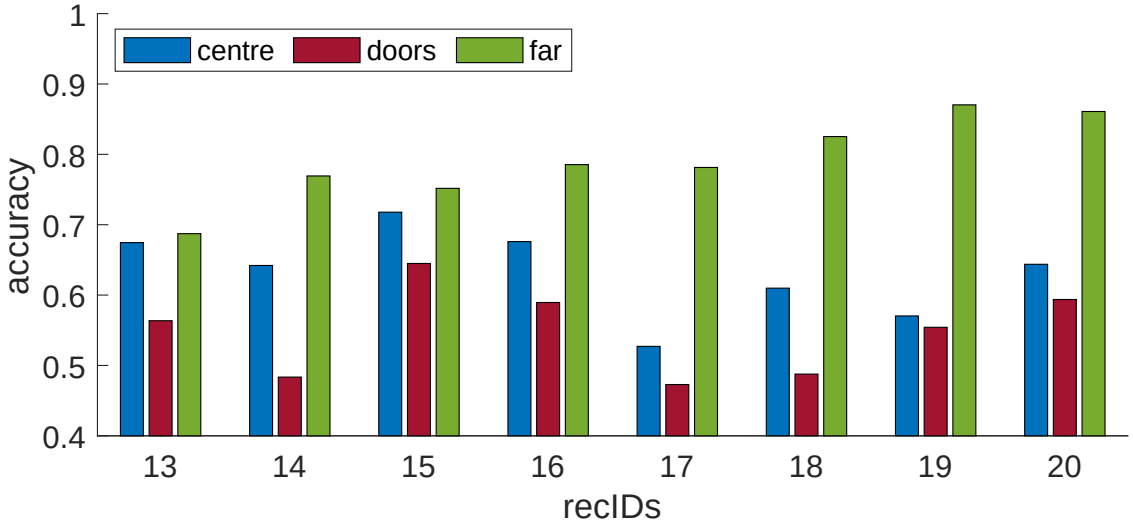
To conclude, with the **Room Detection** step, we have exploited the IoT signals received in the smartwatch as the user freely moves around the house to infer estimates of their initial semantic locations. However, as we have seen, *maxRSSI* can often have limited localisation accuracy, as it exploits the BLE data only and can thus not handle missing data or interferences from nearby rooms effectively. In the next section, we will see how we can use the additional mobility information provided by the smartwatch's IMU data to eventually improve upon the results of *maxRSSI* in Section 4.5.1.

4.5.2 Motion Detection: the *energyPeaks* approach

The WatchBLoc dataset does not log any information about the activities performed during the experiment. As such, a direct data-driven evaluation of the proposed *energyPeaks* approach as a walking event detector is not possible. However, our mobility estimates are supported by prior research on sedentary and active behaviour [359–363]. This section first presents relevant statistics from the literature, followed by an analysis of the experimental results of the *energyPeaks* module.



(a) *demo-home*



(b) *real-home*

Figure 4.3: *maxRSSI* accuracies for all users and beacon configurations

The accuracy of *maxRSSI* is calculated for three different device configurations (*centre*, *doors*, *far*) for each user. Depending on the configuration, we observe significant variability in the *maxRSSI*'s accuracy. Configurations behave similarly for all users in the same environment, i.e., configuration *centre* is best performing in the *demo-home* scenario (left). In contrast, *far* has the highest accuracy in the *real-home* scenario (right). We also observe a large variance across the users within a configuration setup, even within the same environment, particularly for the *demo-home* case. Note that the axes start at 0.4 instead of 0, as *maxRSSI* exceeded this performance in all cases.

As predictions in this work are generated over windows of 5.12; *sec*, all reported statistics will be expressed in this metric to facilitate comparison with estimates derived from *energyPeaks*. The conversion is performed using the following formula:

$$X\ hours = \frac{X\ hours \cdot 60 mins/hour \cdot 60 secs \min}{5.12\ sec/window} \approx ceil(703.125X)\ windows,$$

where X denotes the duration of an event in hours, *ceil* rounds the result up to the nearest integer, and *windows* represents time segments of 5.12 seconds each. Below, statistics from the literature are presented in their original format, followed by their equivalent in windows (e.g., 10 *hours* (7032 *windows*)).

In modern societies, adults spend approximately 62% of their waking hours at home, equating to around 10 hours per day [360] (7032 windows). Research further indicates that 58% of waking hours are spent sedentary, with 39% allocated to light-intensity activity and 3% to vigorous activity [361]. Walking at a moderate pace, typically classified as light-intensity activity [364], is common in home environments. Assuming light-intensity activities are uniformly distributed between home and external settings, approximately $39\% \cdot 10$ hours = 3.9 hours (2743 windows) of light-intensity activity would occur at home. However, this encompasses various activities such as household chores and cooking, not just walking.

To estimate the proportion of at-home light-intensity activity attributable to walking, we consider typical adult walking patterns. While the recommended daily step count is 10,000, actual averages range from 3,000 to 5,000 steps per day [362, 363]. Using a median estimate of 4,000 steps and assuming an even distribution between home and external environments, approximately $62\% \cdot 4000 = 2480$ steps would occur at home. Given an average walking cadence of 80–100 steps per minute [365] (with 90 steps per minute as a representative estimate), this translates to approximately $\frac{2480}{90} \approx 28$ minutes (329 windows) of walking per day at home, accounting for 4.68% of time spent at home.

Finally, studies on room-to-room transitions indicate that healthy adults change rooms approximately 111 times per day [359]. Given that a typical day consists of 10 hours at home, and that only one transition can occur per window, this

results in an estimated $\frac{111}{10;hours} = \frac{111}{7032;windows} \approx 1.58\%$ of time spent at home corresponding to room transitions.

With these in mind, let us inspect the result of the *energyPeaks* module.

- The watchHAR dataset has recorded a total of 45981 *windows* of 5.12 *sec* data.
- 555/45981 windows were annotated as ground truth room transitions, representing $\approx 1.21\%$ of the total time of recordings, which is within acceptable range of the literature-based estimate of 1.58% for the percentage of time spent at home corresponding to room transitions.
- 45426/45981 windows were annotated as ground truth of not-changing-rooms.
- 1895 windows were estimated as walking events by *energyPeaks*; this corresponds to 1895/45981 = 4.12% of total events corresponding to walking, which is within acceptable range of the above literature-based statistic of 4.68% of time spent at home corresponding to walking events.
- 507/1895 identified walking events occurred at the same time as ground-truth room transitions, meaning that only 48 walking events occurring at room transitions were missed, with 505/555 (91.35%) of room transitions being correctly identified as such. By examining the 48 missed walking events around room transitions, we observed that these occured at transitions to adjacent rooms; on such occasions, less vigorous movement is expected, as well as shorter duration for the walking event, as only a short distance needs to be covered. As a result, the accelerometer energy is expected to be lower and shorter in duration, hence occasionally missed by the threshold of *energyPeaks*.

The high accuracy of detected walking events at ground truth room transitions, along with the strong alignment between the estimated total walking events during waking hours and values reported in the literature, supports the effectiveness of *energyPeaks* as a robust method for walking event detection, which is yet simple and largely parameter-free. While *energyPeaks* accurately identifies walking events

near room transitions, it cannot inherently differentiate between walking events associated with room changes and those occurring within a single room change of the user's room-level location. Consequently, to achieve precise indoor positioning, *energyPeaks* must be complemented with additional room occupancy information to accurately determine the user's location within the house.

In the next section, we combine the outcomes of **Room Detection** and **Motion Detection** to provide a final, refined room-level user location.

4.5.3 Room Estimation: inference and parameter learning

4.5.3.1 RoomI: choosing the model parameters carefully

Having initialised the parameters $\theta \equiv (E_R, E_M, T, \pi)$ of the model, we can infer the most likely sequence of underlying states of the model for our observations, using the Viterbi algorithm. The trail of the landing rooms of these states corresponds to our room-level location estimates.

In principle, if the initial parameters of the HMM are carefully chosen, the Viterbi path can give us already improved location estimates compared to the single sensor modality estimates. This is the simple version of our method, which omits the learning step of the HMM; we denote this special case of **RoomE** as Room Inference (**RoomI**). Fig. 4.4 (second group of boxplots) summarises the performance of **RoomI** for the three different device configurations; in comparison with the **Room Detection** results from *maxRSSI* (leftmost group of boxplots), **RoomI** demonstrates an improved performance, averaging to 80.46%.

Fig. 4.5b revisits the recording we studied in Fig. 4.5a. **RoomI** effectively fuses the location data with the mobility data to account for the no-signal phases of *maxRSSI* and significantly improves on the adjacent rooms' noise. We will see how we can further improve this behaviour by learning the parameters of the HMM to allow for a more informed final decision on semantic localisation.

4.5.3.2 RoomE: learning the model parameters

Following the training of these parameters, we perform inference on the learnt model, thus obtaining the final room-level location estimates generated by our method. Our method can consider two inference function modes: *offline* and *real-time* inference. In the *offline* mode, the model computes the Viterbi path based on historical data (e.g., the full recording of the day) and then reports the full trail of estimated semantic locations of the user. In the *real-time* mode, the model computes the Viterbi using the L most recent observations ($L = 12$ in our case) and reports the current user location as the final room of the current Viterbi path estimate. Fig. 4.4 summarises the performance of the *maxRSSI*, **RoomI** and **RoomE** estimates. **RoomE** achieves an average performance of 81.53%, an improvement of 20.3% on the *maxRSSI* estimates. **RoomE-RT**, denoting the *real-time* mode of our method, achieves an overall 77.6% localisation accuracy. As is apparent in the rightmost group of boxplots in Fig. 4.4, the *real-time* performance can be almost as good as the *offline* performance, depending on the device configuration. Fig. 4.5c demonstrates the improvement of **RoomE** on the recording we examined in Fig. 4.5a and Fig. 4.5b.

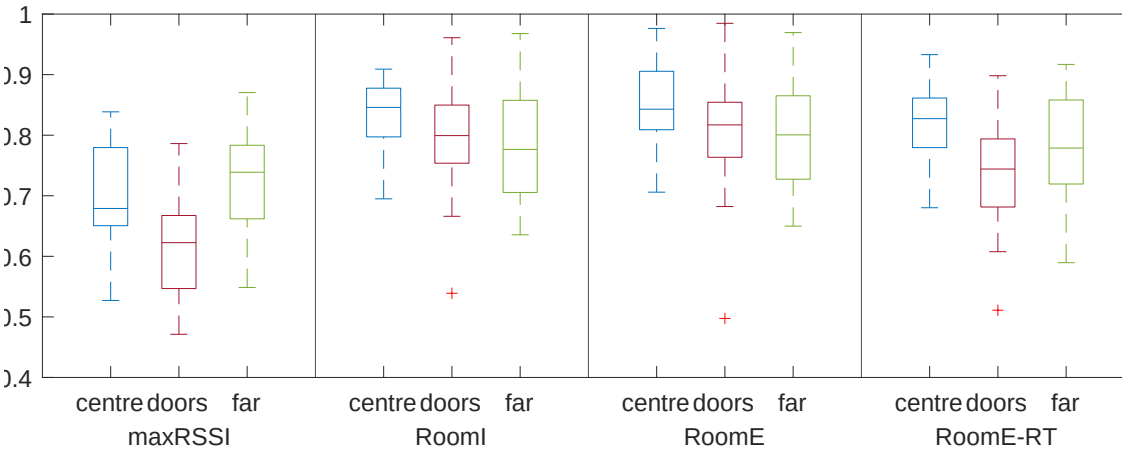
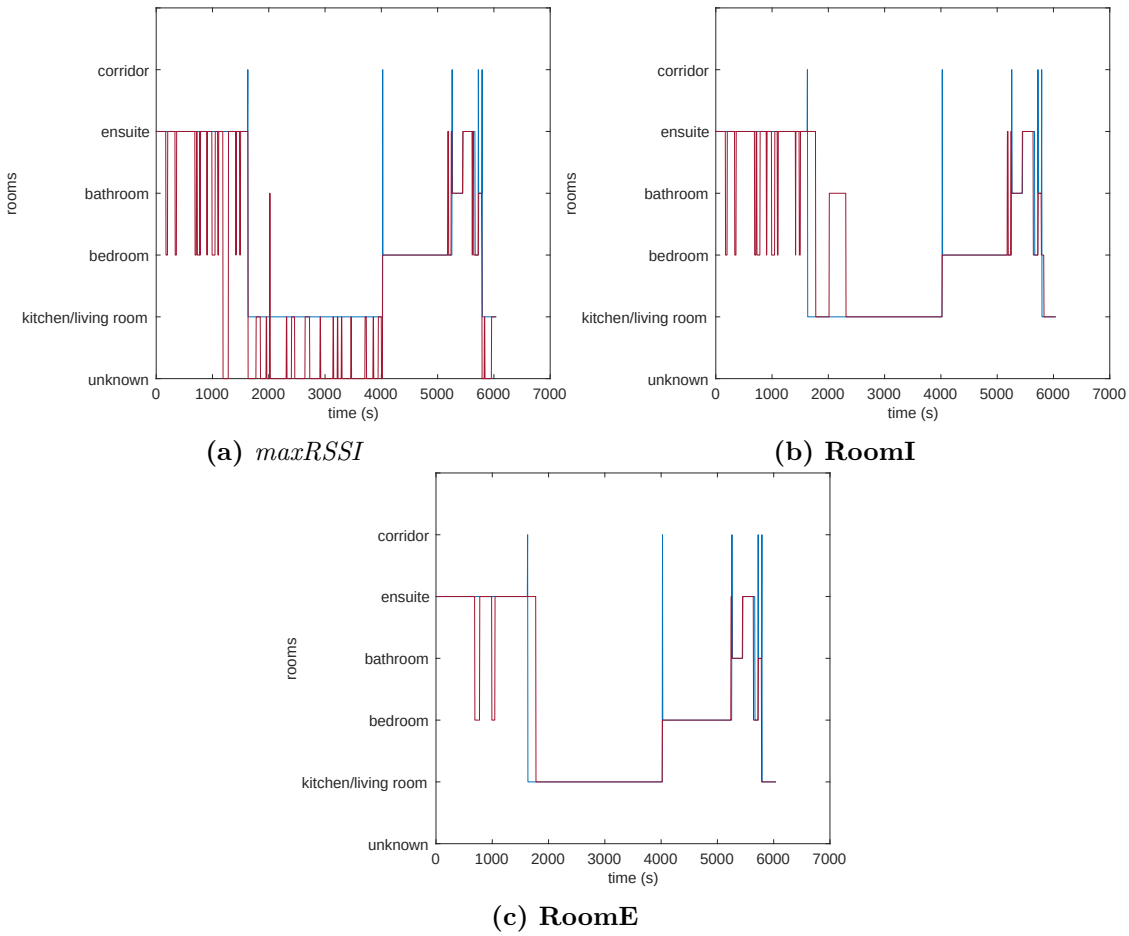


Figure 4.4: Overall statistics

The boxplots for the *centre*, *doors* and *far* configurations demonstrate the improved performance of **RoomI** compared to *maxRSSI* case, and the further improvement of **RoomE** against both **RoomI** and *maxRSSI*. The **RoomE-RT** section demonstrates the performance of **RoomE** in real-time. Note that the axes start at 0.4 instead of 0, as all methods exceeded this performance.



(a) *maxRSSI* **(b) RoomI**

(c) RoomE

Figure 4.5: Estimated vs ground truth locations for recID 19 : *centre*.

(a) The *maxRSSI* fails to identify areas where the strongest signal comes from an adjacent room to the one the user is currently in. This is often the case for the adjacent `ensuite` and `bedroom` rooms of the *real-home*. For areas where there is no IoT device installed (e.g., corridors) or when no beacon is heard (e.g., in some parts of the *kitchen/living room*), the *maxRSSI* is unable to provide a location estimate. (b) **RoomI** effectively smooths out the previously unknown areas and improves the noise from adjacent rooms but is still prone to interferences. (c) **RoomE** further improves on the noise from adjacent rooms.

4.5.3.3 Semantic map estimation

With the learnt model, we can exploit the learnt state transition matrix to extract information about room connectivity, yielding a connectivity graph that resembles a semantic map of the house.

Each node in the connectivity graph represents a room within the house, while each edge denotes a transition between two rooms. The thickness of an edge

is proportional to the probability of transitioning between the respective rooms. If behavioural patterns are embedded within the connectivity graph, as will be discussed below, this allows for directed edges; in that case, each pair of rooms in the graph should be connected by two directed edges, representing movement in both directions.

We propose two approaches for generating the connectivity graph from the state transition matrix: "a behavioural-based approach", which preserves statistics regarding room occupancy and mobility patterns, and a "privacy-based approach", which removes behavioural information and simplifies the graph to reflect only room connectivity.

The initial steps of the process are common to both approaches.

From state-transition matrix to room-transition matrix Recall that the Hidden Markov Model (HMM) produces a learned state-transition matrix, where each state encodes a room transition. This state-transition matrix is normalised in the standard manner, ensuring that each row sums to 1.

To derive the room-transition matrix from the state-transition matrix, we must determine the likelihood of being in a state that represents a particular room transition. Consider an illustrative example of a house with two rooms, r_A and r_B. The output state-transition matrix of the HMM is structured as shown in Table 4.1.

Table 4.1: Example of a state-transition matrix for a house with two rooms, r_A and r_B

	$(r_A \to r_A)$	$(r_A \to r_B)$	$(r_B \to r_A)$	$(r_B \to r_B)$
$(r_A \to r_A)$	a	b	0	0
$(r_A \to r_B)$	0	0	c	d
$(r_B \to r_A)$	e	f	0	0
$(r_B \to r_B)$	0	0	g	h

where a, b, c, d, e, f, g, h are probabilities such that $a + b = 1$, $c + d = 1$, $e + f = 1$ and $g + h = 1$. Zero values indicate impossible transitions, as a valid transition requires the final room in the starting state to match the initial room in the arriving state.

To construct the room-transition matrix, we consider the number of valid ways to arrive at each state. For instance, the state $(r_A \rightarrow r_A)$ can be reached from both $(r_A \rightarrow r_A)$ and $(r_A \rightarrow r_B)$. Since all states are defined symmetrically, the number of valid transitions remains the same for all states. Consequently, summing each column provides a representative *score* for each room transition, as shown in Table 4.2. These scores are not probabilities but are comparable, as they result from the same number of valid transitions.

Table 4.2: Example of scores for each state, generated from the state-transition matrix for a house with two rooms, r_A and r_B

room transition	$(r_A \rightarrow r_A)$	$(r_A \rightarrow r_B)$	$(r_B \rightarrow r_A)$	$(r_B \rightarrow r_B)$
score	a + e	b + f	c + g	d + h

By reshaping the calculated score vector, we obtain the unnormalised room-transition matrix, as shown in Table 4.3. Standard row-wise normalisation of this matrix ensures that each row sums to 1, thereby producing the room-adjacency matrix, which serves as the foundation for constructing the graph-based semantic map of the house.

Table 4.3: Example of the generated unnormalised room-transition matrix for a house with two rooms, r_A and r_B

	r_A	r_B
r_A	a + e	b + f
r_B	c + g	d + h

Behavioural-based normalisation This "behavioural-based normalisation" of the room-adjacency matrix retains information beyond room connectivity, incorporating: i) room occupancy patterns (i.e., the probability of remaining in the same room), ii) movement habits (e.g., frequency of transitions between specific rooms). Such behavioural insights are particularly valuable in applications such as remote monitoring of older adults, where deviations from usual mobility patterns may indicate potential health concerns or the need for intervention.

Privacy-based normalisation In some cases, preserving behavioural patterns may pose privacy concerns, necessitating their removal from the generated map. To achieve this, we propose a privacy-based normalisation, which involves two key steps:

1. Eliminating directional movement biases: The probability of transitioning between two rooms is made symmetric, removing information about habitual room-visit sequences. This is done by averaging the scores for transitions in both directions: $score_{(r_A \to r_B)} = score_{(r_B \to r_A)} = \frac{b+f+c+g}{2}$.

2. Removing room occupancy information: The probability of staying within a room is set to a uniform value across all rooms: $score_{(r_A \to r_A)} = score_{(r_B \to r_B)} = \frac{a+e+d+h}{2}$

Following this, the score vector is reshaped into a matrix and normalised row-wise, as described in the behavioural-based normalisation approach. The resulting graph-based semantic map now reveals only room connectivity without behavioural details. In this case, the graph does not require directed edges, as a single undirected edge suffices to indicate connectivity between two rooms.

Comparison with ground truth adjacency matrix Once the room-adjacency matrix is established, it must be compared with the ground truth adjacency matrix of the environment. The ground truth matrix is typically a binary adjacency matrix, which is not directly comparable to the real-valued room-transition matrix. To facilitate comparison, both matrices must be renormalised as follows:

- Ground truth matrix scaling:

    1. A probability p is assigned to room occupancy (i.e., remaining in the same room); this is the same for all rooms.

    2. The remaining probability $1 - p$ is evenly distributed among all valid room transitions from each room. This ensures each row sums to 1. Table 4.4 illustrates this for our two-room example.

- Room-transition matrix scaling:

    1. Probabilities of staying within a room are set to p, as above.
    2. Remaining transition probabilities in each row are scaled to sum to $1-p$, ensuring each row sums to 1.

For evaluation, we use $p = 50\%$ but the exact choice of p does not affect relative comparisons between the matrices. The key metric remains the difference between the rescaled ground truth matrix and the room-transition matrix, rather than their absolute values.

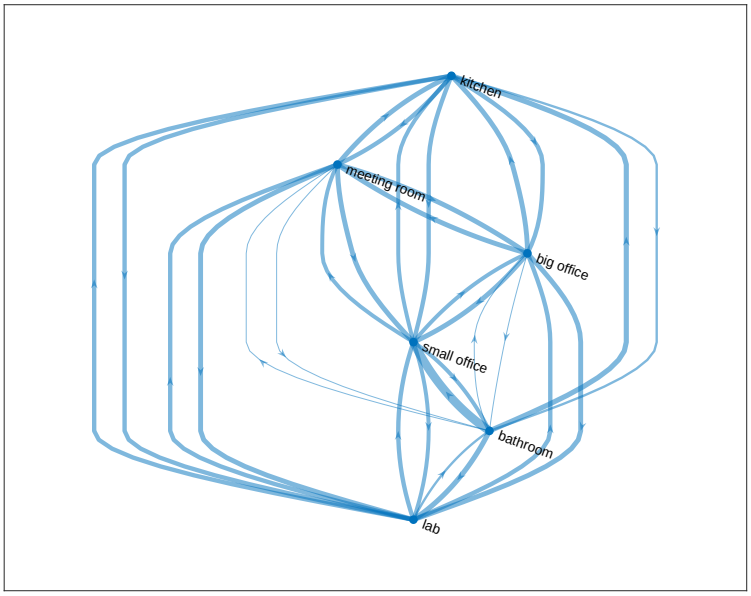
	r_A	r_B
r_A	p	1-p
r_B	1-p	p

Table 4.4: Example of the generated unnormalised room-transition matrix for a house with two rooms, r_A and r_B

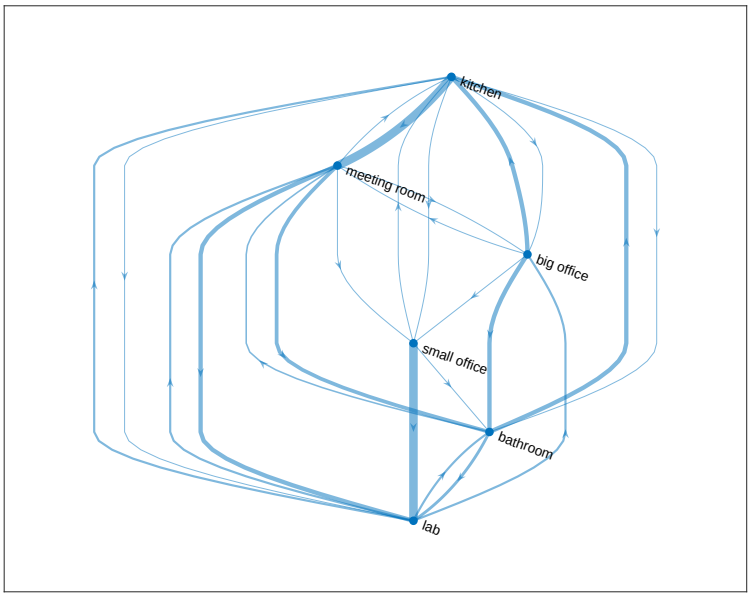
The average accuracy of the generated semantic maps is 97.85%, denoting the effectiveness our proposed methodology to generate an adjacency matrix that is representative of the environment dynamics. An original and estimated "behavioural-based" map of the *demo-home* environment are shown in Fig. 4.6; We observe that with our proposed approach, the generated graph-like semantic map correctly identifies transitions across rooms; with some edges being thicker, it captures higher-frequency transitions over less common ones. It also captures further behavioural patterns; for example, we observe transitions from and to the same room to occur at difference frequencies, which is true for the "demo-home" environment that corresponds to an office-space with large corridors interconnecting all rooms.

4.6 Related work

Human positioning has been a prominent research area for decades, with estimates ranging from accurate spatial coordinates (*geographic localisation*) to semantic location (*semantic localisation*) or even orientation of the pose of the individual (*pose estimation*). While GPS provides accurate outdoor tracking, indoor positioning



(a) Real map.



(b) Estimated map.

Figure 4.6: Semantic maps.

The ground truth connectivity graph of the *demo-home* environment is compared to an estimated connectivity graph generated with our method. The estimated map resembles the original map regarding connectivity and transition probabilities between rooms. Some information regarding the frequency of visiting each room (i.e., the room occupancy statistics for the user) is still encoded in the estimated connectivity graph, leading to some missing edges, e.g., from `bathroom` to `big office`. These correspond to low-probability transitions in the original graph and do not cause a significant error in our estimation.

presents challenges due to signal attenuation, infrastructure costs, and privacy concerns. This section focuses on research relevant to our proposed approach, RoomE, highlighting methods that are lightweight, cost-effective, and privacy-preserving. For a broader discussion on camera-, sound-, and light-based positioning, as well as further RF-based techniques such as UWB localisation, which, despite their accuracy, are unsuitable for private indoor environments like the ones we examine in this chapter, we refer the reader to Section 2.4.3.

RF-based localisation In the area of RF-based localisation, Wi-Fi and BLE have been the most commonly employed sensor modalities that ensure lightweight infrastructure and user privacy. Wi-Fi positioning typically uses trilateration or fingerprinting, with the latter being influenced by signal propagation models that are difficult to maintain in indoor environments. Device dependence and computational complexity in large spaces are also challenges. Solutions include local propagation models and device-independent fingerprint maps using statistical analysis and weighted k-NN. With high energy consumption, Wi-Fi is increasingly being replaced by Bluetooth Low Energy (BLE) beacons in indoor localisation applications due to BLE's power efficiency [89, 124, 135, 136, 139, 140]. RoomE also follows this approach, employing BLE as a key sensor modality for positioning.

Compared to Wi-Fi, BLE offers a stronger correlation between Received Signal Strength Indicator (RSSI) and distance, leading to improved localisation accuracy [139, 140]. Consequently, various Wi-Fi and BLE-based indoor positioning approaches have emerged, with fingerprinting being a widely adopted method [29, 137, 138, 141, 142]. While numerous techniques aim to address challenges such as fingerprint map size and device dependency, most still require an environmental map [143] and a well-defined path-loss model [144]. However, obtaining such data can be particularly challenging in private indoor spaces, such as residential settings. RoomE overcomes the need for fingerprinting and environment maps, by effectively fusing proximity information to BLE beacons with simple mobility estimates.

Inertial navigation systems The integration of MEMS sensors into smartphones, tablets, and smartwatches, has given access to the collection of mobility data via accelerometers, gyroscopes, and magnetometers, enabling Pedestrian Dead Reckoning (PDR) [138, 152]. While inertial data can be noisy, due to how devices are held and used, leading to sensor, walking, and alignment errors, IMU-based positioning remains central to indoor positioning research due to the widespread availability of wearable devices. PDR methods include strapdown navigation and step-and-heading estimation.

Strapdown navigation affixes an IMU to a tracked body, estimating position via double integration of acceleration and angular velocity [154–156]. However, integration errors cause drift, commonly mitigated by Zero-Velocity Updates (ZUPT), where foot-mounted IMUs reset velocity during the stance phase of human gait [155, 158].

Step-and-heading estimation reduces error propagation to two dimensions by detecting steps and heading changes but is affected by stride length variations [157]. RF-based localisation is often integrated, using smartphone Wi-Fi and inertial signals for pedestrian tracking [160]. BLE beacons dynamically switch between triangulation and fingerprinting, achieving 1–1.5 m accuracy [161]. Alternative approaches incorporate ultrasound time-of-arrival calculations with a maximum a posteriori (MAP) algorithm to refine positioning [162]. However, the step and heading technique also faces issues with varying stride lengths, which introduce further errors in location estimation. These factors complicate the accuracy and reliability of INS.

Hybrid and semantic localisation The fusion of ambient RF signals and inertial data has been effective in indoor public spaces equipped with extensive wireless sensor networks, typically using Wi-Fi APs or BLE beacons alongside smartphone IMU sensors to achieve positioning accuracy within 2–10 metres [29, 30]. However, these methods struggle in private indoor spaces, where large positioning errors may place a user in an entirely different room. Moreover, they often rely on extensive fingerprinting or crowdsourcing [175], which is impractical in home settings. While

smartphones enable positioning in public environments, they are not consistently carried indoors, limiting continuous and accurate tracking.

While the aforementioned methods predominantly focus on geographical localisation, semantic localisation is often more relevant for home environments, as it preserves privacy by eliminating the need for personal attributes such as human height (to estimate step size) or detailed floorplans (for map matching and fingerprinting-based approaches). In this context, [171] introduce a method leveraging human-robot coexistence and smartphone-based HAR to estimate room-level location. Rather than externally recording locations, BLE beacons were deployed, and ground truth was assigned based on the beacon emitting the strongest RSSI signal at any given time. However, the use of robots limits real-world applicability of this method. A notable human-centric and infrastructure-light approach is S-Smart [172], which integrates HAR with PDR to dynamically infer household locations such as windows and doors. However, it requires an extensive on-body sensor setup, including a pocketed smartphone, a wrist-worn IMU, and a foot-mounted sensor, making it impractical for daily use. Similarly, AtLAS [173] combines HAR and PDR from smartphone IMU data with Wi-Fi RSSIs to determine user location and identify landmarks, though it relies on Wi-Fi fingerprinting. In contrast, [174], following [366], base their approach solely on HAR from smartwatch IMU data and BLE RSSIs from room-installed devices, using supervised learning to establish a relationship between RSSI values and room-level locations. While this is the most similar approach to RoomE, it remains very sensitive to the fusion mechanisms.

RoomE Our proposed approach utilises a single wearable smartwatch sensor and ambient BLE signals, restricting required infrastructure to everyday devices. It eliminates the need for fingerprinting or environment mapping by integrating both sensor modalities and employs Hidden Markov Models for sensor fusion, effectively capturing the temporal dependencies of events detected by each modality.

4.7 Conclusions

In this chapter, we introduced **RoomE**, a sensor fusion method acting on ambient BLE and inertial data to achieve privacy-preserving indoor localisation. We evaluated the method on the **WatchBLoc** dataset, a comprehensive dataset of ambient BLE and smartwatch IMU data for seamless room-level localisation that we introduced in Chapter 3.4. We discussed the challenges in such positioning scenarios and demonstrated behaviours across different users, environments and device placements. Our method is less variable to device configuration compared to the state-of-the-art methods and achieves accurate semantic localisation and mapping for the user at home.

This work has focused on gaining insights into a user's behaviour by examining their interaction with their environment, delving into the semantic understanding of that environment, and analysing how they utilize different spaces by moving between rooms and determining the duration of their presence in each *semantic location*. However, understanding daily behavioural patterns entails unravelling not only the intricacies of the user's surroundings but also the user's dynamic interactions within various contexts. In the following chapter, we will further investigate human behaviour analysis, specifically focusing on the identification of distinct human activities.

Different semantic locations often trigger a diverse range of activities, each with its own significance in the given semantic context. For instance, spending time in the bedroom typically signifies relaxation or sleep, while frequent movement within the bedroom implies a distinct set of activities, such as preparing to depart or tidying up, among others. As such, recognising the user's current activity serves as a valuable means to distinguish between semantic locations that may have a similar likelihood of occurrence. Given our imperative for precise and real-time monitoring, acquiring activity information at the requisite level of granularity becomes pivotal. For instance, while identifying an energetic activity can provide insights that reduce ambiguity regarding whether the user is in the bedroom or at a different location, capturing finer details such as whether the user is brushing their teeth or washing

dishes (both considered energetic activities but providing more specific context) can offer the critical distinction between the bathroom and the kitchen, respectively.

To this end, the following chapter introduces a hierarchical approach to activity recognition that aims to capture activities at multiple granularity levels and provide informed decisions across diverse semantic contexts.

Chapter 5

Hierarchical activity recognition for activities of daily living

5.1 Introduction

In the previous chapter, we explored infrastructure-free, privacy preserving solutions to estimating a user's semantic location within a private indoor space. To further enhance our understanding of human behaviour, we will investigate in this chapter how to efficiently and concurrently provide activity information to relevant mobile applications, that may require knowledge of a user's activity patterns for their operation, further exploring the area of human behavioural understanding.

5.1.1 Motivation and challenges in human activity recognition

Human activity recognition (HAR) is an active area of research in ubiquitous computing. Providing appropriate and concise information about the activities people perform in their everyday lives, commonly known as activities of daily living (ADL), is helpful in various application domains, such as remote healthcare, localisation, smart homes and workout monitoring [367, 368].

In the last few years, various studies have focused on the problem of identification of human activities using a plethora of sensors, ranging from wearable devices capturing motion to cameras, ambient sensors (e.g., WiFi) and even acoustic sensors [39, 369]. Given the advances in computer vision, camera-based approaches, in particular, have been prevalent in solving the HAR problem. However, image retrieval poses significant privacy concerns for the users involved, making camera usage, and thus also the corresponding techniques, prohibitive for many scenarios, e.g., remote monitoring of older adults.

On the other hand, wearable motion-capturing sensors are free from such constraints. The increasing smartphone usage and their increasing sensing and computing capabilities have shifted the focus of HAR research to methods that use a smartphone as the primary sensing device [37–39]. Some approaches rely solely on smartphone-based inertial data (acceleration, gyroscope, magnetometer). Others employ a fusion of inertial data with other sensor modalities that phones capture, e.g., WiFi [370–372].

Smartphones, however, can also pose significant constraints in HAR in free-living conditions. Most HAR algorithms are developed and evaluated in a controlled environment. However, experiments in controlled environments often ignore the intermittent use of smartphones, particularly in indoor environments. Even though most people will hold their phone or place it in a bag or pocket while outdoors, it is not common to carry the device constantly when, e.g., moving around the house or the office. Instead, the device will most commonly lie on a nearby surface while the person goes ahead with everyday indoor activities. As such, smartphone-based HAR methods are often unable to operate as an assistive tool for, e.g., remote monitoring of the elderly, as smartphones cannot provide the uninterrupted sensing required in such scenarios.

At the same time, even when their intermittent use is not an issue, smartphone-based approaches are often constrained by the placement of the mobile device. Smartphones have effectively been used to tackle a range of interesting activities (walking, running, sitting, lying, etc.), and the effect of different phone placements

has been well-studied [31, 37]. However, it is not straightforward to identify activities of primarily manual dexterity (e.g., a handshake or teeth-brushing) using smartphone recordings solely [373].

A separate challenge of the HAR task arises from the varying demands of the ever-increasing mobile applications that could benefit from an accurate estimate of the user's ADL. Each such application has been developed to assist or monitor a different aspect of everyday life and thus may require 1) information about a different set of activities, 2) different levels of detail in the activities of interest, or 3) information about a specific activity only intermittently, compared to other applications that also run on the device. Furthermore, most mobile applications require accurate information fetching and processing in real-time, which can be challenging under the limited computational resources of mobile devices.

Based on the above, it becomes evident that a HAR system that can support simultaneously a multitude of mobile applications and improve situation awareness in each application's scope should have the following properties: 1) be able to provide each application with the exact information it requires and at the level of detail requested, and 2) minimise the computational energy in the mobile device.

5.1.2 Proposed approach and key contributions

In this work, we will focus on using smartwatches to solve the problem of HAR for ADL while overcoming the difficulties that arise from using smartphones. Smartwatches are wrist-attached and thus carried by the bearer of the device wherever they go. The wrist placement also enables the identification of more complex activities, such as washing or cooking, that involve vigorous hand movements that may be missed by a smartphone placed in, e.g., the person's pocket [373]. We propose **CHiHAR**, a *centralised, hierarchical* approach to *HAR* that estimates ADL from smartphone IMU data, setting the basis for a framework that can robustly and efficiently support the demands of a variety of different applications.

Our proposed approach stems from the observation that the HAR problem for the ADL scenario involves a plethora of complex tasks, e.g., "washing", "eating",

"walking", etc., which are, however, at the same time highly granular: each task can be broken down into more detailed individual activities or can be itself part of a higher-level, more generic activity. For example, reaching for a knife and fork, cutting a piece of steak, directing the fork to the mouth, and placing the cutlery down may all provide essential information for applications that monitor, e.g., fine motor skills for rehabilitation after a stroke. For other applications, though, e.g., a healthy lifestyle calendar, the higher-level activity "eating" composed of the above-detailed activities will be informative enough.

Our hierarchical framework identifies activities initially at a high level (e.g., "walking on stairs") and then, if necessary by the use case, refines these high-level classes into more granular ones (e.g., "walking upstairs", "walking downstairs"). It employs a combination of feature engineering, primarily based on the wavelet transform, and latent feature extraction through a CNN-LSTM-based architecture effectively fused to learn the characteristics of the tasks involved at each granularity level. We build and evaluate our proposed approach on **watchHAR**, the dataset of smartwatch-based sensor data and associated granular ADL information, introduced in Chapter 3.4.

In summary, we make the following contributions:

- We identify the modular property of ADL that allows them to be naturally grouped based on their logical or context-based similarities or be broken down into more detailed components, as and when required by the demands of each mobile application installed on the user's mobile device.

- We propose **CHiHAR**, a centralised hierarchical framework that achieves robust classification performance for various ADLs at a low computational cost over a hierarchy of decision levels. Our method is easily deployed as it only requires a wrist-attached smartwatch without any requirements for calibration or user constraints.

- We evaluate a prototype of our proposed system on the **watchHAR** dataset, introduced in Chapter 3.4. We also inspect its inference-time capabilities on

three devices of varying computing power: a workstation server and two mobile devices (a tablet and a phone). Our proposed approach consistently improved the average F1 score performance across classes with Leave-One-Subject-Out evaluation from 73.16% to 76.67% on average and reduced up to 4× the computational cost, compared to the non-hierarchical HAR benchmark.

The remainder of this chapter is organised as follows. Section 5.2 presents the architecture of our proposed method and illustrates how the different modules work together. Section 5.3.2.2 introduces the algorithmic details for each of the system's components and explains how this aids HAR both during training and real-time inference. Section 5.4 prepares the data we will use in this work, and Section 5.5.2.2 then evaluates the performance of our proposed method. Finally, Section 5.7 discusses limitations and future directions.

5.2 System architecture

This section provides a high-level description of our system architecture and its main algorithmic components.

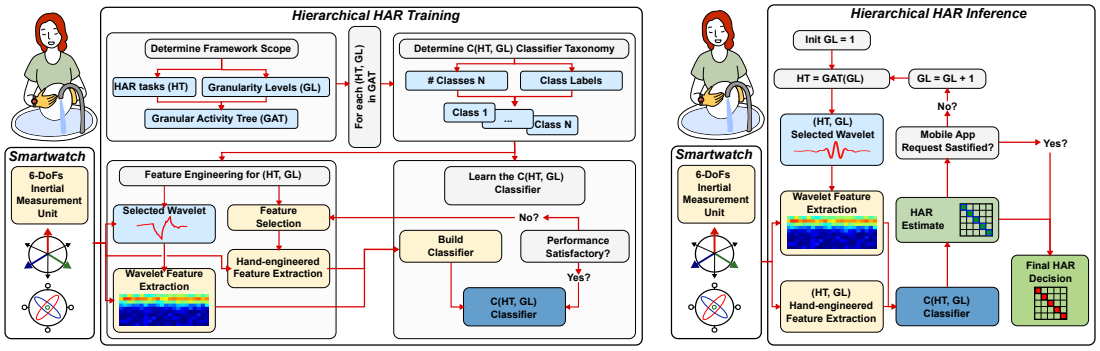
Our algorithm aims to accurately identify both simple and complex ADL and provide fast response times regarding the activity performed without any initial calibration of the smartwatch or any constraints to the user's movements. This is achieved through two main components, depicted in Fig. 5.1 and outlined below:

1. **hierarchical HAR learning** to analyse HAR tasks across multiple scales and provide a fast and informative estimate on the activity performed across N hierarchical levels of learning.

2. **feature engineering**, consisting of:

   (a) **wavelet feature extraction** from the raw data using an appropriately selected mother wavelet to extract informative time-frequency features from the signal,

(b) further **hand-engineered feature extraction** for each level in the hierarchy of learning to assist robust learning of individual recognition tasks.



(a) Training. **(b)** Inference.

Figure 5.1: System Architecture for **CHiHAR**.

Our framework comprises of separate training and inference modes. During training, the details of the hierarchical framework need to be established. Following a feature engineering module for each level of chosen semantic abstraction, we train the classifiers and inspect their performance. At inference level, the trained hierarchical models are sequentially accessed (from more abstracted to more detailed) until estimates at the required level of semantic detail are acquired.

The hierarchical HAR learning module is the backbone of our proposed approach, assisted by the remaining two feature-based components that operate under its modular scope. It enables the synthesis and breakdown of activities from low- to high-level task descriptions and effectively learns to distinguish them at various granularity levels.

The two feature engineering submodules are designed to perform a semi-automated feature extraction. In particular, we estimate selected features from the IMU data to transform the information in the raw data into two representative sets of features.

- The first contains pattern-based features in the form of informative 2D scalogram images captured by the wavelet feature extraction module. These visualise the energy patterns of the signals across time, enabling the use of existing deep architectures, e.g., CNN-based models for image recognition,

while maintaining the sequential nature of the IMU data. An example scalogram can be seen in Fig. 5.2, in comparison with the raw triaxial accelerometer data.

- The second set of features is strongly coupled with the hierarchy of HAR learning; at each level in the hierarchy, it selects and extracts the features that best strengthen the identification of the task under investigation. This allows us to build robust classifiers with more targeted scopes that can operate as a swarm in any combination required by the task of interest.

In summary, our hierarchical framework operates as follows. Several specific-to-task classifiers are built and trained to encode different levels of detail for each activity in each subclassifier. Then, at inference time, the IMU input recordings travel along the hierarchy of trained classifiers to estimate the activity performed by the user at the scale of detail requested at any given time.

5.3 Algorithms

In this section, we will discuss the details of our proposed approach. We will begin by introducing the hierarchical learning framework, followed by the feature engineering method under the hierarchical framework.

5.3.1 Hierarchical learning framework

In our proposed hierarchical framework, we encode multiple levels for various activities to achieve high granularity and low computational cost. We design a swarm of classifiers; we train each classifier to recognise a high-level activity group, a subgroup within the high-level group, and the detailed activities inside the subgroup. For example, we build three classifiers to identify a detailed activity, such as “wash hands”, which belongs to the “manual tasks” group and the “personal hygiene” subgroup.

In the remainder of this section, we will first define the hierarchical framework’s scope and present the details of training the swarm of classifiers. Then, we will

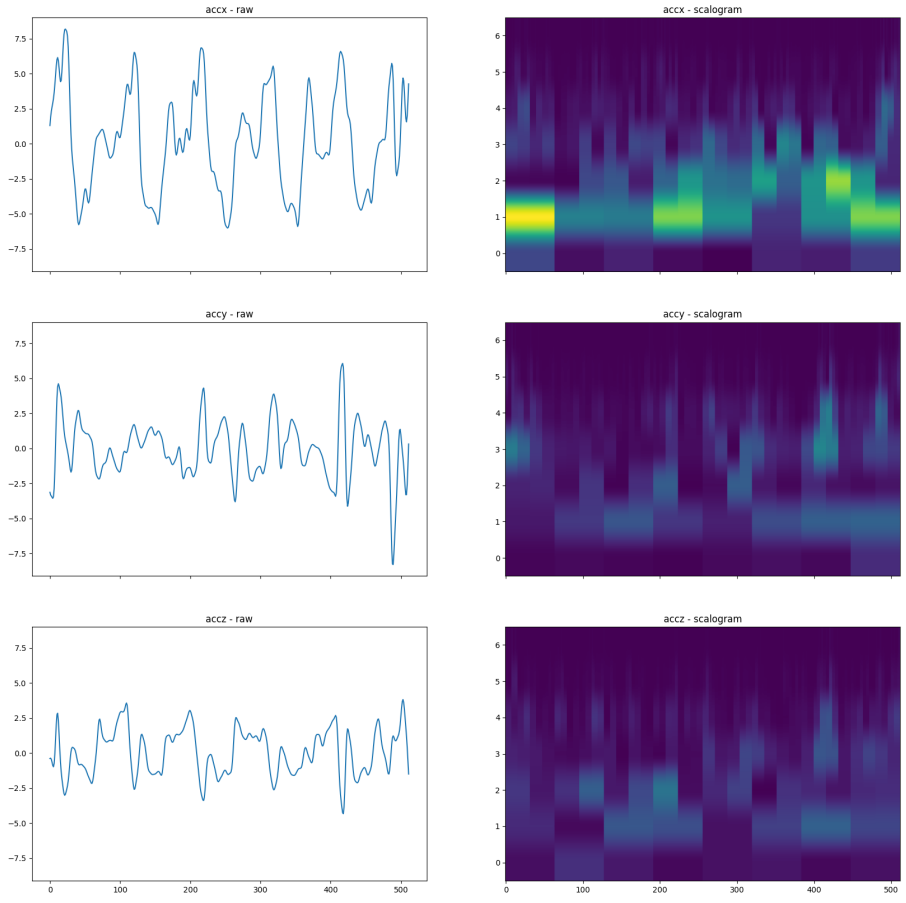


Figure 5.2: Example of scalograms generated from raw accelerometer data.

Visual representation of scalograms for each axis of a sample accelerometer data recording. The scalograms retain the signal patterns by depicting its energy over time, providing a visual representation that informatively captures changes in the data both over time and in terms of the signal's energy content.

explain how these collaboratively work at inference time to provide a final informed decision on the activity performed.

5.3.1.1 Hierarchical HAR training

Determine the framework scope To train our swarm of modular classifiers, we first need insight into the types of mobile applications that require activity information at some level of granularity. There are three key directives that we need from each application: i) the application context, i.e., what is the main goal of each application; ii) the list of activities that are relevant in each application to achieve that goal; and, iii) the application focus, meaning whether detailed information regarding how the activity is performed is necessary, or whether information on the occurence or not of the activity suffices.

Let us demonstrate these key components through an illustrative example. Assume five mobile applications that are taken into account to build the hierarchical HAR system, with the following contexts: "physical activity monitoring", "eating habits", "personal care goals", "work habits". The relevant activities for these applications are as follows:

- "physical activity monitoring": "walk", "relax", "exercise"
- "eating habits": "eat", "drink", "prepare snack", "prepare meal"
- "personal care goals": "wash face", "brush teeth", "brush hair", "shower", "exercise", "cook"
- "work habits": "write", "type"

For each of these, it is necessary to identify the application's goal; for example, regarding the "brush teeth" activity, does the application require information about how well the activity the user is brushing their teeth (e.g., to ensure good dental hygiene), or does the information whether the teeth are brushed or not suffice (e.g., to monitor the mental health status of people suffering from depression by measuring their adherence to key acts of personal care). For activities where the

way of execution is required, a precise list of subactions should be obtained, defining the underlying rules that define the different styles of execution is required (e.g., poor/good teeth brushing could be based on optimal number of brushing strokes or brushing directions to be completed).

As can be seen, activities requested by different applications may overlap either directly, e.g., as is the case for the activity "exercise" above, or in a broader context, e.g., "prepare snack" and "prepare meal" are all acts of cooking.

Identifying such contextual similarities among the activities is key to building the hierarchical system. However, there is a lot of flexibility on how activities can be linked together to form different contexts; in some cases, for instance, considering the example of quality of teeth brushing, there is a direct linkage of activities; in this example, the set of movements that define the quality of teeth brushing are more granular activities of the higher-level activity class "brush teeth". In other cases, however, merging activities under a common coarser class label is not as obvious; in this case, how concepts are linked together, e.g., manual grouping, using ontology mechanisms or language models to automatically identify connecting concepts among activities, is key to the generated hierarchy. Despite their differences, the output of any of these approaches is dependent on the focus the hierarchical framework is called to support, as this is decided by the framework's designer. For example, valid approaches could include: separating activities as indoor and outdoor, grouping activities together when they commonly occur in a specific room of the house, or by their movement-type similarity (e.g., locomotive activities and activities involving manual dexterity). In this work, we utilise manual hierarchy generation, grouping activities based on their movement-type similarities.

Once these choices have been made, and the conceptual similarities of activities have been established based on the framework designer's selected method, a tree-like hierarchy is constructed to convey coarse-to-detailed information about the activities involved in a top-down way. We define this hierarchy of labels as the Granular Activity Tree (GAT). This is the basis of our hierarchical framework. Fig. 5.3 provides an illustrative example of a simple GAT.

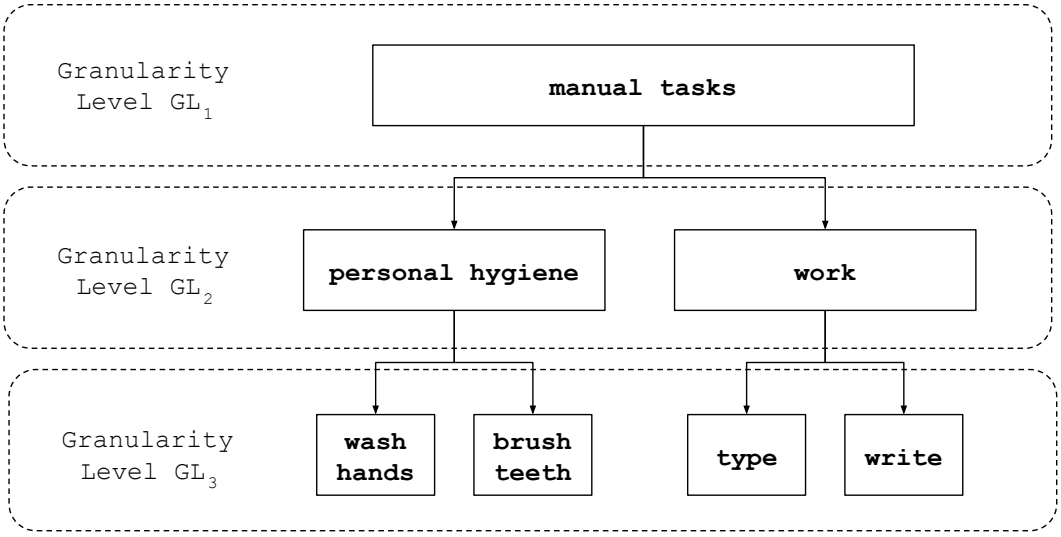


Figure 5.3: Example of hierarchy generation in a small branch of a Granular Activity Tree (GAT).

This is an illustrative example of hierarchy generation, demonstrating the selection of activity labels at each level of granularity. The example focuses on a small branch of a Generalised Activity Tree (GAT) to clarify the hierarchical classification process, which can be generalised for constructing the full tree. We examine the following scenario: i) The hierarchical framework is designed to support multiple applications, two of which focus on monitoring work habits and pre-bedtime hygiene routines. Consequently, distinguishing between these two categories is essential in the GAT design, hence a granularity level should include the categories "personal hygiene" and "work", which must be differentiable by the classifier corresponding to this GL. ii) The work-focused application examines the user's preference between writing and typing. As more specific than "work", these belong to a deeper granularity level and should be identifiable by the subclassifier that is activated when a "work" task occurs. iii) The hygiene-focused application monitors young children's bedtime routines to ensure they develop consistent habits, such as washing their hands and brushing their teeth before bed. Similar to the "work" subclassifier, a "personal hygiene" subclassifier should distinguish between these two activities, which, as more detailed actions, also reside at a deeper GL. iv) The applications do not require information about how these activities are performed (e.g., distinguishing between proper and insufficient hand-washing techniques). Consequently, there is no need for a further detailed granularity level beyond the ones identified. v) Further applications supported by the framework do not require specific details about activities that exclusively involve hand movements (which is the case for the work and hygiend activities). Therefore, all these activities can be grouped under a higher-level category, "manual tasks", allowing the classification system to focus only on activities relevant to hand-executed tasks. As the most generic, this should lie at the most coarse granularity level. Note that this example assumes the existence of additional classes (e.g., "idle") at the first granularity level (GL_1). However, to maintain simplicity in the visual representation, these additional classes are not depicted in the sample.

Determine the classifiers' taxonomy We then build separate classifiers for each granularity level (GL) in the GAT. Activities that lie in the same GL in the GAT (except for the most detailed GL, the GL_{max}) form a "HAR task" (HT); this means that we need to build a tailored classifier to distinguish between the activities in the HT. We will denote this classifier as $C(HT, GL)$. For GL_{max}, a HT is formed by each group of activities in GL_{max} that also share a common parent. We define the $C(HT, GL)$ classifier's scope, i.e., its output classes, as the activities in HT.

Let us illustrate the above through an example, based on the breakdown to GL depicted in Fig. 5.3. GL_1 corresponds to the most abstract activity label and thus sets the nature of the types of activities the classifier in GL_2 should expect to decide upon. Then, a binary classifier for GL_2 will aim to solve the HT for the classes "personal hygiene" and "work". GL_3 corresponds to GL_{max}; thus we build a subclassifier for each group of leaves in GL_{max} that share a parent in GL_2, namely: the subclassifier $C(personal\ hygiene, GL_3)$ with output classes "wash hands" vs "brush teeth", and the subclassifier $C(work, GL_3)$ with scope "type" vs "write".

Having defined the scope of our framework and the details of the hierarchy of classifiers, we are now able to discuss the classifiers' input features, followed by the HAR learning details.

Feature engineering for classifiers The feature engineering accounts for two sets of features: time-frequency features, based on the discrete wavelet transform (DWT) in the form of a 2D scalogram image, and other hand-engineered features, selected appropriately for each $C(HT, GL)$.

Wavelet transforms have proved their effectiveness in HAR from inertial data [209–212]. As a time-frequency representation, the wavelet transform preserves the sequential nature of inertial data, by also highlighting the changes in energy content over time within a recording. It also ensures higher resolution than other frequency representations, e.g., STFT [374]. At the same time, the use of scalograms transforms inertial data into images, allowing for the use of convolutional-based

deep architectures for the IMU-based HAR task, which have proved highly effective in image classification problems.

Pivotal in this work, is that the hierarchical framework enables the choice of different mother wavelets for each classifier, tayloring the wavelet choice to optimally match the key discriminative patterns in the time-frequency domain within each classification task, which is not possible for the flat classification approach, where a single wavelet has to be utilised. We investigate the effect of wavelet choice in Section 5.5.2.2.

The choice of hand-engineered features is also flexible in the proposed hierarchical approach, allowing each $C(HT, GL)$ to be built upon modules that best assist in solving the HT in question.

A number of approaches can be utilised to define and select the hand-crafted features. Statistical features that disregard domain knowledge, such as signal mean and standard deviations, distribution-based features for compact signal representations or learned features directly from raw data, commonly utlising deep models, can be utilised based on what information and computational resources are available to the framework's designer [39, 65, 375, 376]. While manual feature engineering is often prone to error and may lack generalisability across domains [375], it has been shown to be valuable when integrated in the feature engineering and selection process [377], to accurately capture the key aspects of the problem at hand, leading to better model performance in that domain [65].

To this end, as we will discuss in Section 5.5.2.2, a feature that aids classification in one problem may not necessarily support another similar problem. For example, in this thesis, we utilise a gravity feature as a proof of concept of hierarchical feature engineering. Gravity is selected as an appropriate feature based on domain expertise: as it can be used as a loose device orientation estimate, it can support pose identification, which is relevant for the C_{walk} subclassifier we define as part of the experimental evaluation of this work. Indeed, the addition of gravity as a supporting feature improves classification accuracy for this task.

However, as demonstrated in Section 5.5.2.2, introducing the same gravity feature in $C(coarse_{HAR}, GL_{min})$ does not aid the classifier's performance, as pose is not a key discriminative factor for that classification problem.

Training the classifiers With the scope of each $C(HT, GL)$ defined and the two sets of features selected and extracted, we can build all the $C(HT, GL)$ classifiers based on the following simple principles:

- All classifiers are formed around a basic 2-layer ConvLSTM architecture that expects 2D scalograms as inputs; these are in the form of images with 6 channels, one for each channel of IMU sensor data (namely, 3-axial acceleration and 3-axial gyroscope).
- Each additional hand-crafted feature in $f(HT, GL)$ is concurrently passed to a separate CNN module that operates in parallel with the primary CNN module processing the wavelet features.
- The latent feature representations from all CNN modules are fused to a single latent vector before going through two sequential LSTM modules.
- A final Fully-Connected layer, followed by a SoftMax layer, provides the final $C(HT, GL)$ model's output.

Fig. 5.4 provides a sample visualisation of the classifiers. After training each classifier independently based on the framework's scope and taxonomy, our system is ready to provide HAR estimates at the GLs defined in the GAT.

5.3.1.2 Hierarchical HAR inference

In this section, we will discuss the inference process of our proposed hierarchical framework for two different query types: first, we will show how the system operates for tasks of the type "which activity is currently performed in GL?", $GL \in [GL_{min}, GL_{max}]$. We will call these *exploratory tasks*. Then, we will discuss the inference process for the query type "is activity X currently performed?", $X \in GAT$, which we will call *awareness tasks*.

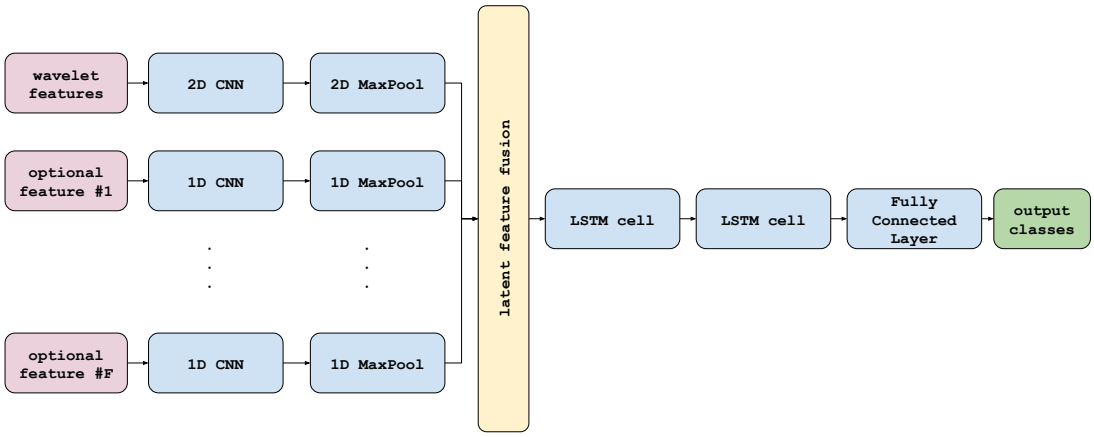


Figure 5.4: Prototype architectures for all classifiers.

All classifiers are based on a 2-layer ConvLSTM architecture, with 2D wavelet scalograms as inputs. In the hierarchical framework, each subclassifier is tailored with additional optional hand-engineered features, processed by CNN modules parallel to the main Conv2D module. A flat classifier equivalent to the hierarchical framework should contain all optional features utilised in the various subclassifiers.

To solve an *exploratory task*, we first extract from the logged IMU data the features required by the highest-level classifier and input the extracted features to the high-level classification model $C(HT, GL_{min})$. If a more detailed GL is required from the *exploratory task*, we extract any further features needed by the 2$^{\text{nd}}$ level classifier $C(HT_2, GL_2)$, and obtain the HAR estimate for GL_2. We repeat the process until the required GL is reached and report on our final informed decision about which activity is performed at the requested granularity level. Note that only one activity can be selected at each level of the hierarchy.

The process is similar for *awareness tasks*; in this case, the inference process begins as described above but stops at the first GL in which the current HAR estimate is no longer a superset of the requested activity. E.g., revisiting Fig. 5.3, suppose an application wants to know whether the user is currently "typing"; the IMU recordings are processed and passed to the "personal hygiene" vs. "work" classifier, and the estimated HAR is "personal hygiene". However, this is not a predecessor of "type" in the labels hierarchy; thus, we do not need to further explore in deeper GL.

In this section, we have presented the end-to-end learning and inference process for our proposed hierarchical learning framework. Next, we will introduce the theoretical and implementation details of the feature engineering module and further discuss how it operates under our hierarchical framework for HAR.

5.3.2 Feature engineering under hierarchical framework

With the hierarchical HAR framework fully defined, we are now in a position to present the algorithmic details for the feature engineering module and highlight how the standard hand-engineering of features is adapted as part of the hierarchical scope of our proposed system.

The feature engineering module includes two main components: wavelet feature extraction and extraction of other hand-crafted features from smartwatch IMU data. The first submodule targets a *pattern-based feature extraction*, while the final one is tailored to best enhance the HT in each subclassifier $C(HT, GL)$.

5.3.2.1 Wavelet feature extraction

With an appropriate choice of wavelet for the task of interest, we apply the wavelet transform on all channels of the raw data, namely triaxial acceleration and triaxial gyroscope. The output of the wavelet transform is a set of coefficients for each wavelet decomposition level applied. We then represent the calculated coefficients as a 2D scalogram with 6 channels (one for each channel of IMU data), i.e., a visual representation of the coefficient energies at different levels across time, which we will use as input to each subclassifier $C(HT, GL)$.

5.3.2.2 Hand-engineered feature extraction

The above-described wavelet feature extraction is a necessary component of our proposed hierarchical learning framework. Here, we will discuss how a separate set of optional features, selected appropriately for each HT, further supports hierarchical learning during the training process and at inference times.

For each classifier $C(HT, GL)$ assigned to a specific classification problem, we assume two modules to be in place: first, a module that suggests features that

could assist the learning of the current classification task. Second, a module that selects a subset of the proposed features that best improve the performance for the $C(HT, GL)$ classifier.

In principle, the fusion of many hand-engineered and latent features is often advised in classification tasks to assist the learning process in deep architectures [74, 75]. This method fits and is realised in our framework; however, our proposed approach of selective feature engineering per level of the hierarchy further ensures that features are extracted from the raw data only when considered necessary by the current HT. We thus build each classifier $C(HT, GL)$ only from components that will assist its task of interest. Instead of creating a complex classifier that will have to balance the information from possibly a multitude of hand-engineered features, we build a set of simpler classifiers, each tailored to solve a smaller classification task. This "lazy" approach to feature engineering also reduces the computational energy during inference time; a feature that is selected in the highest level of abstraction, i.e., for $C(HT, GL_{min})$, and then needed again further down the hierarchy, can be computed once at the beginning of the inference process and reused when necessary. In contrast, a feature at an intermediate GL that is only required to solve a single HT will only be calculated if the inference process has to reach the corresponding $C(HT, GL)$ to answer upon an exploratory or awareness task.

Our proposed feature-engineering approach effectively supports the introduced hierarchical learning framework by simplifying learning for all intermediate tasks. It can also reduce the computational energy at inference time compared to non-hierarchical classification approaches, as we will demonstrate in Section 5.5.2.2.

5.4 Data preparation

In this work, we evaluate on the watchHAR and the WISDM datasets.

A basic preprocessing is considered necessary to prepare the data for our hierarchical framework. Similarly to the IMU recordings of the **WatchBLoc** dataset, discussed in Section 4.4.2, we first account for any missing data with linear interpolation. We then proceed to denoise the signal with a moving median filter.

Next, we separate the gravity component using a high-pass Butterworth filter with a cut-off frequency of 0.3 Hz. As the frequencies of human activities are limited to 20Hz, we then use a low-pass Butterworth filter with a cut-off frequency of 20 Hz to remove any high-frequency noise. The data are then bounded, normalised to zero mean and unit standard deviation, and windowed to $5.12sec$ windows with 75% overlap.

5.5 Evaluation

In this section, we will discuss the experimental details of our proposed approach. We will evaluate a prototype of the hierarchical framework over its classification accuracy performance and inference capabilities compared to the standard non-hierarchical approach to HAR.

Methodology We employ the newly introduced dataset **watchHAR** and the WISDM dataset to train and evaluate the prototype model and benchmarks in terms of classification accuracy and inference times.

For both datasets, the hierarchy design is determined manually, according to motion-based similarities of the recorded activities. In both scenarios, we identify a subset of walking activities, that lead to the user changing their location; a subset of idle activities, where the user remains static and in calmness; and a subset of manual activities, where the user is static but engages in activities that demand manual dexterity.

We perform leave-one-subject-out evaluation for all subjects in each dataset, utilising the left-out subject as the test set. From the remaining users, 80% of each user's data are utilised for training, and the remaining 20% for validation. The watchHAR dataset contains 21 subjects, and WISDM 51 subjects. For the watchHAR dataset, we utilise both the accelerometer and gyroscope recordings. For the WISDM dataset, we utilise only the smartwatch accelerometer recordings. All data are preprocessed as discussed in Section 5.4.

Precision, recall and f1-score, calculated per class and as average values are utilised to inspect classification accuracy. Time elapsed in milliseconds is utilised to measure the inference time capabilities of the model.

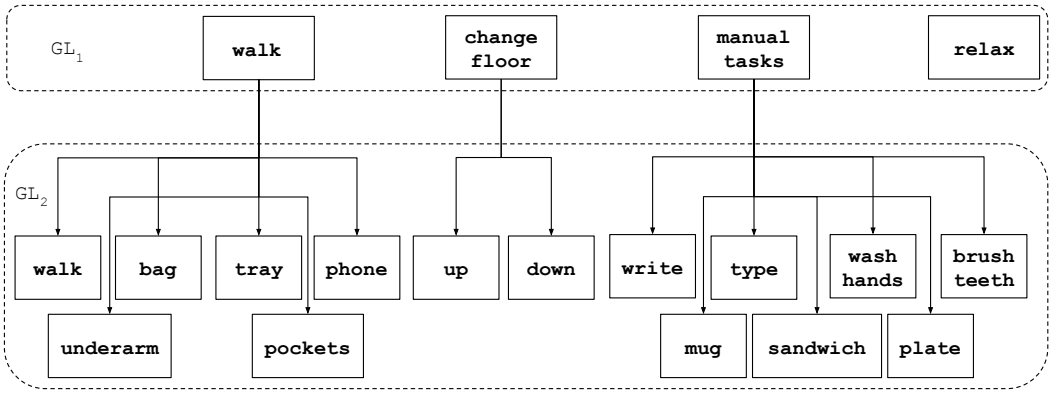
5.5.1 Hierarchical HAR learning

5.5.1.1 The prototype framework

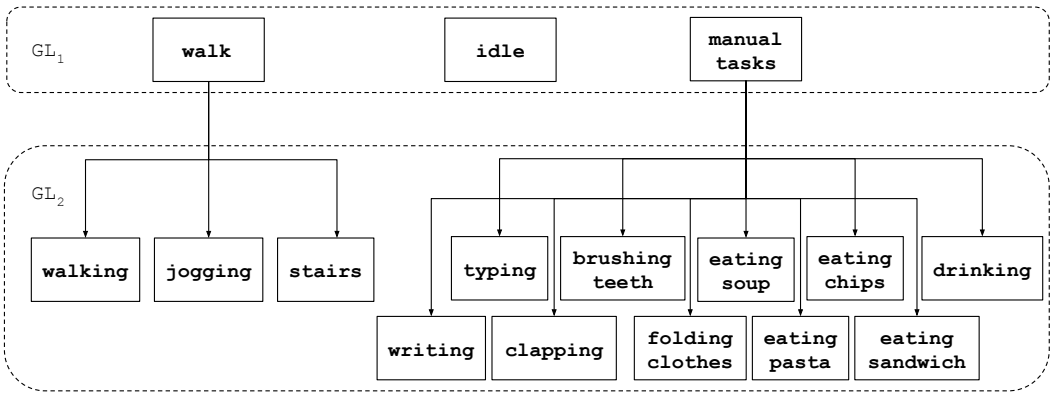
To demonstrate our proposed hierarchy of learning ADL, we build a prototype swarm of classifiers. For both the watchHAR and WISDM datasets, we define a hierarchy over two levels.

For both datasets, the hierarchy design is determined manually, according to motion-based similarities of the recorded activities. In both scenarios, we identify a subset of walking activities, that lead to the user changing their location; a subset of idle activities, where the user remains static and in calmness; and a subset of manual activities, where the user is static but engages in activities that demand manual dexterity. Note that for the WISDM dataset, we consider only a subset of activities, excluding samples labelled as "Kicking", "Catch", and "Dribbling". This decision ensures greater alignment between the two datasets by prioritising at-home activity patterns and reflects the fact that these predominantly foot-based activities are unlikely to be meaningfully captured by smartwatch inertial data. Also, the activities "Standing" as "Sitting are jointly considered as the activity "idle". The GAT for the two datasets, depicted in Fig. 5.5a, 5.5b, respectively, thus determine the scope of the hierarchical training in each domain.

The watchHAR classifiers Based on the GAT in Fig. 5.5a, our prototype consists of the following four classifiers: a $coarse_{HAR}$ classifier $C(coarse_{HAR}, GL_1)$ at GL_1 to categorise among the high-level descriptions of the activities in our dataset. Then, three subclassifiers, one for each group of activities in GL_2 that share a parent, namely: $C(walk, GL_2)$ with six output classes in its taxonomy, $C(change\ floor, GL_2)$ with two output classes and $C(manual\ tasks, GL_2)$ with



(a) Granular Activity Tree for watchHAR.



(b) Granular Activity Tree for WISDM.

seven output classes. Note that the high-level activity "relax" does not have any children in the GAT; thus, no subclassifier for that activity is required.

The WISDM classifiers Similarly for the WISDM dataset, based on the GAT in Fig. 5.5b, our prototype consists of the following four classifiers: a $coarse_{HAR}$ classifier $C(coarse_{HAR}, GL_1)$ at GL_1 to categorise among the high-level descriptions of the activities in our dataset. Then, two subclassifiers, one for each group of activities in GL_2 that share a parent, namely: $C(walk, GL_2)$ with three output classes in its taxonomy, and $C(manual\ tasks, GL_2)$ with ten output classes. Note that the high-level activity "idle" does not have any children in the GAT; thus, no subclassifier for that activity is required.

For the remainder of this chapter, we will denote the hierarchical framework consisting of the above-defined classifiers as C_{hier}.

We select the `bior2.2` mother wavelet across all levels of the hierarchy, as it has successfully been used in prior work [213]. In terms of additional hand-engineered features, a gravity feature is considered, loosely hinting at the smartwatch's orientation during the activity.

Gravity, also often known as acceleration bias, is always embedded in the accelerometer signal, which comprises of three key components: gravitational influence, movement-induced acceleration, and noise [378]. The gravity force $\overrightarrow{g_F}$ corresponds to the acceleration occurring due to gravity; assuming a static accelerometer, and a level frame with its z-axis in the same direction as gravity, the gravitational acceleration would be:

$$\overrightarrow{g_F} = [0, 0, g],$$

where $g = 9.81 m/s^2$. To identify its effect on triaxial acceleration data, let us express the accelerometer vector $\overrightarrow{a} = [a_x, a_y, a_z]$ in a frame tilted at roll (ϕ), pitch (θ) and yaw (ψ) angles. This is a common representation for the accelerometer data, as tilt can be captured by inertial measurement units.

$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} cos\phi & 0 & -sin\theta \\ sin\theta sin\phi & cos\phi & cos\theta sin\phi \\ sin\theta cos\phi & -sin\phi & cos\theta cos\phi \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ -g \end{pmatrix} \Rightarrow$$

$$\Rightarrow \begin{cases} a_x = gsin\theta \\ a_y = -gsin\phi cos\theta \\ a_z = -gcos\phi cos\theta \end{cases}$$

As can be seen by the equations above, gravity affects the values of triaxial acceleration. As a wrist-worn accelerometer moves along with the user's movements, the accelerometer frame is rotated with respect to the level gravity frame. As such, identifying the direction of gravity can provide regarding device orientation. This can be valuable information for activity recognition, particularly for tasks that aim to identify different poses of the same activity [379], e.g., standard walking and walking while talking on the phone.

To identify gravity, we filter the accelerometer signal using a high-pass Butterworth filter with a cut-off frequency of 0.3 Hz [379, 380].

While not the only way to estimate device orientation [31, 381, 382], estimation of the gravity vector can be achieved using simple methods, such as signal filtering, and can be accessed directly through accelerometer data. Hence, we prefer it for its simplicity and ease of estimation.

As we will discuss in more detail in Section 5.5.2, the gravity feature was selected only for the $C(walk, GL_2)$ classifier, which focuses on identifying different poses of walking, as the user may be carrying different objects held at angles that yield different smartwatch orientation; the rest operate solely with their extracted wavelet features. The remaining classifiers could also be augmented with further domain-appropriate features for each; here, we only focus on the effect of feature utilisation as a proof-of-concept, hence constrain our evaluation to a single feature (gravity) and a single classifier, namely $C(walk, GL_2)$.

In summary, we build and train the following classifiers:

- Each of the $C(coarse_{HAR}, GL_1)$, $C(change\ floor, GL_2)$ and $C(manual\ tasks, GL_2)$ classifiers for the watchHAR dataset are composed by:
  - a 2D convolutional layer, followed by a 2D max-pooling layer, then
  - a 2-layer LSTM cell and
  - a fully connected layer (FC) (followed by a SoftMax layer) as the output layer.
- The $C(walk, GL_2)$ classifier in watchHAR and all the WISDM classifiers are built with:
  - a 2D convolutional layer (followed by 2D max-pooling), to process the wavelet features,
  - a 1D convolutional layer (followed by 1D max-pooling), parallel to the 2D convolutional layer above, to process the gravity features,
  - a latent feature fusion layer to fuse the outputs of the above convolutional modules, followed by

  - a 2-layer LSTM cell and
  - a FC and SoftMax layer, as above.

Finally, to demonstrate the benefits of our hierarchical approach, we also build and train the corresponding non-hierarchical classifier for the above tasks; for the remaining of this work, we will denote the non-hierarchical classification as *flat* classification and the corresponding classifier as C_{flat}. C_{flat} should contain all the individual components identified in the swarm of hierarchical classifiers, namely the wavelet features and the full set of additional hand-engineered features that have been selected across the various levels of hierarchy in our proposed approach, i.e., the gravity feature in our prototype.

As such, the C_{flat} classifier for our investigated hierarchical prototype consists of:

- a 2D convolutional layer (followed by 2D max-pooling), to process the wavelet features,
- a 1D convolutional layer (followed by 1D max-pooling), parallel to the 2D convolutional layer above, to process the gravity features,
- a latent feature fusion layer to fuse the outputs of the above convolutional modules, followed by
- a 2-layer LSTM cell and
- a FC and SoftMax layer, as above.

The hyperparameters for each model were chosen individually through extensive hyperparameter search and are summarised in Table 5.1.

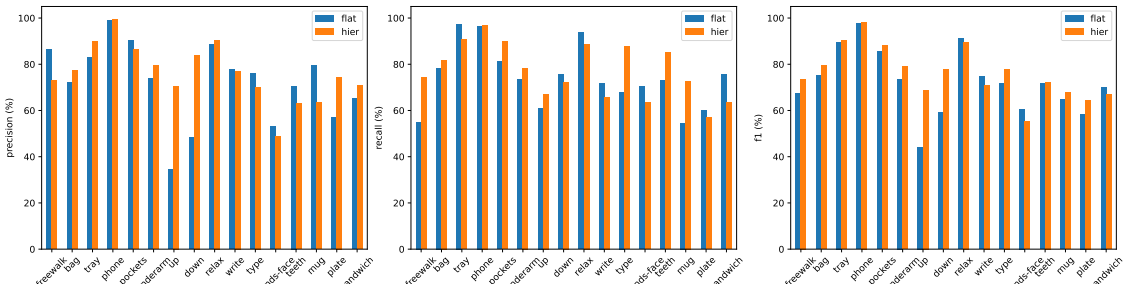
5.5.1.2 Classification accuracy

To effectively evaluate the classification performance of our proposed C_{hier} compared to the non-hierarchical C_{flat}, we need to consider the performance of the two models under the same scope. In our framework, this is equivalent to identifying each activity at its most refined granularity level, GL_{max}, i.e., the set of 16 detailed

Table 5.1: Hyperparameter values table.

Hyperparameter	C_{coarse_{HAR},GL_1}	C_{walk,GL_2}	$C_{change\ floor,GL_2}$	$C_{manual\ task,GL_2}$	C_{flat}
learning rate	10^{-4}	10^{-6}	10^{-4}	10^{-5}	10^{-6}
epochs	100	800	100	1500	1500
Conv1D	-	inputChannels=3	-	-	inputChannels=3
	-	inputSize=(1,512)	-	-	inputSize=(1,512)
	-	kernelSize=3	-	-	kernelSize=3
	-	padding=1	-	-	padding=1
	-	stride=1	-	-	stride=1
MaxPool1D	-	kernelSize=3	-	-	kernelSize=3
	-	padding=0	-	-	padding=0
	-	stride=3	-	-	stride=3
Conv2D	inputChannels=6, inputSize=(5, 512), kernelSize=(3,3), padding=1, stride=1				
MaxPool2D	kernelSize=(3,3)	kernelSize=(3,3)	kernelSize=(1,3)	kernelSize=(1,3)	kernelSize=(1,3)
	padding=0, stride=kernelSize				
other	bs=128, winSize=512, Conv{1D, 2D}-ouputChannels=96, LSTM-hidden=96				

activities described in Table 3.2 of Chapter 3.4. We evaluate the two methods on our collected dataset, using Leave-One-Subject-Out evaluation, to best investigate the ability of the models to generalise in the lack of calibration procedures. Fig. 5.6 reports the average precision, recall and F1-score across the 21 subjects of the **watchHAR** dataset for each of the 16 activities of interest.



(a) Precisions. **(b)** Recalls. **(c)** F1 score

Figure 5.6: Classification accuracy comparisons between C_{flat} and C_{hier}.

Our proposed C_{hier} consistently improves the average F1-scores for 12 out of the 16 activities and ameliorates the overall average HAR performance measured by three accuracy metrics, summarised in Table 5.2 for both datasets.

Table 5.2: Average HAR accuracy.

	watchHAR		**WISDM**	
Accuracy metric	C_{flat}	C_{hier}	C_{flat}	C_{hier}
average precision	72.21%	**76.12%**	57.03%	**57.39%**
average recall	74.14%	**77.22%**	**61.37%**	60.92%
average F1-score	73.16%	**76.67%**	61.38%	**64.65%**

Although the WISDM dataset exhibits lower overall performance, the hierarchical approach still enhances classification accuracy, albeit with a smaller margin compared to the watchHAR dataset. This discrepancy may stem from the decision to use only accelerometer data for WISDM, whereas the watchHAR dataset benefits from the fusion of accelerometer and gyroscope data, which is known to improve recognition accuracy.

A key advantage of the hierarchical approach is its ability to achieve accurate classification at the highest level of the hierarchy by effectively distinguishing

between groups of samples that exhibit substantial differences. At finer granularity levels, the approach refines classification by identifying more subtle variations among data points that share key similarities, having already been grouped under the same coarse category.

To illustrate these benefits, in Table 5.3 we present the performance of individual classifiers for the WISDM dataset, alongside the confusion matrices for flat and hierarchical classification, shown in Fig. 5.7. The results at the higher levels of classification reinforce our previous discussion. As observed in the confusion matrices, the flat classifier frequently misclassifies the activity "clapping" as "stairs", assigning the incorrect label to the majority of its samples. While the hierarchical approach still struggles to correctly classify "clapping" data, it achieves a more meaningful distinction by categorising the majority of samples under "manual" activities, specifically "teeth brushing". The tendency to confuse clapping with teeth brushing, rather than other manual activities, is particularly noteworthy, as both involve high-energy, repetitive hand movements.

Table 5.3: Average accuracy for the WISDM subclassifiers.

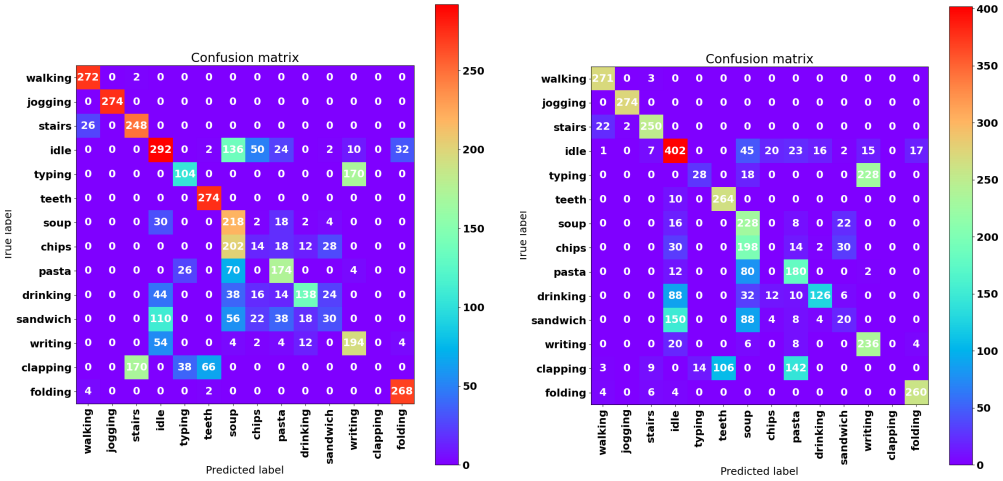
Accuracy metric	C_{coarse}	C_{walk}	C_{manual}
average precision	82.01%	96.66%	47.01%
average recall	86.83%	95.85%	54.12%
average F1-score	83.92%	96.32%	57.99%

In this section, we have demonstrated that the proposed C_{hier} approach to HAR for ADL achieves tangible improvement in the classification accuracy across all three accuracy metrics. In the following section, we will evaluate the model's capabilities in terms of computational efficiency through its estimated inference times.

5.5.1.3 Inference times

Measuring inference times is a key feature to examine, particularly for models expected to operate in-the-wild or run in devices of limited computing power.

To respond to both *exploratory* and *awareness* tasks, C_{flat} needs to run the inference process continually, thus yielding a constant computational cost irrespective



(a) Confusion matrix for C_{flat}, WISDM. **(b)** Confusion matrix for C_{hier}, WISDM

Figure 5.7: Comparison of flat vs hierarchical classification for the WISDM dataset.

of the activity performed by the user. In contrast, our proposed C_{hier} framework can benefit from its modular nature to only enable the subset of subclassifiers in the hierarchy that are required to estimate the task performed at each point in time. The only layer that must continually run the inference process is the highest level in the hierarchy, i.e., $C(coarse_{HAR}, GL_1)$. Any lower-level classifier is activated only when the query has not been effectively responded to by the current HAR estimate, as described in Section 5.2 and depicted in Fig. 5.1.

Intuitively, it can be expected that C_{hier} has longer inference times compared to C_{flat}; it involves the activation of multiple classifiers sequentially, in contrast to the single inference incurred at flat classification. However, there are two key elements that affect the inference time capabilities of the flat and hierarchical frameworks: i) the frequency of subclassifier activations and ii) our definition of C_{flat} as the classification model that includes all optional features encountered in the various subclassifiers.

Inspecting the frequency of subclassifier activations Let N_R be the number of activity information requests generated by *exploratory* and *awareness* tasks. To effectively respond to these requests, the flat classifier C_{flat} needs to be activated

N_R times, yielding a total time:

$$TT_{flat} = N_R T_{flat},$$

where TT_{flat} is the total time C_{flat} is active and T_{flat} is its average inference time.

In the hierarchical framework, the high-level classifier will also have to be activated N_R times, yielding a total time:

$$TT_{coarse} = N_R T_{coarse},$$

where TT_{coarse} is the total time C_{coarse} is active and T_{coarse} is its average inference time. However, a subclassifier at lower granularity levels is activated only when its scope is necessary to fulfil an *exploratory* or *awareness* task. Let A be the set of all possible detailed activities across the various granularity levels of the hierarchical framework and $N_{R(a)} \leq N_R$ be the number of requests a subclassifier receives. Then, each detailed subclassifier $C(a),\ a \in A$ will be activated only $N_{R(a)}$ times, giving a probability of activation $p(a) \in [0, 1]$, with $p(a) = \frac{N_{R(a)}}{N_R}$. Consequently, we derive that for each subclassifier, the total inference time is:

$$TT_a = N_R T_{coarse} + N_{R(a)} T_{C_a} = N_R T_{coarse} + p_a N_R T_{C_a},\ a \in A,$$

hence, the total inference time for the hierarchical framework to execute N_R activity requests is:

$$TT_{hier} = N_R T_{coarse} + \sum_a p_a N_R T_{C(a)},\ a \in A,$$

which for our prototype framework, introduced in Section 5.5.1.1, becomes:

$$\begin{aligned} TT_{hier} &= N_R(T_{coarse} + p_{walk} T_{C(walk)} + p_{stairs} T_{C(stairs)} + p_{manual} T_{C(manual)}) \\ &= N_R T_{coarse} + \sum_a p_a N_R T_{C(a)},\ a \in A \equiv \{walk,\ stairs,\ manual\}. \end{aligned}$$

As the number of requests N_R is common across both the flat and the hierarchical classifiers, for simplicity, we only study the average times:

$$T_{flat} = \frac{TT_{flat}}{N_R}$$

and

$$T_{hier} = \frac{TT_{hier}}{N_R} = T_{coarse} + \sum_a p_a N_R T_{C(a)},\ a \in A \equiv \{walk,\ stairs,\ manual\}.$$

Inspecting the time complexities of each classifier Recall our definitions of C_{flat}, C_{coarse} and $C(a),\ a \in A \equiv \{walk, stairs, manual\}$, as first introduced in Section 5.5.1.1. All models have the same backbone of a 2D ConvLSTM module, with additional parallel 1D convolutional modules to process each hand-engineered feature that has been selected, as illustrated in Fig. 5.4. C_{flat}, in particular, must contain all the components necessary across all subclassifiers to be considered equivalent to the hierarchical framework. Hence, the inference time T_C for a classifier $C \in \{C_{flat}, C_{coarse}, C(a),\ a \in A \equiv \{walk, stairs, manual\}\}$ will be:

$$T_C = \max(T_{CNN(f)}) + T_{LSTM} + T_{FC},$$
$$f \in \{wavelet, optional\ feature\ i,\ i \in \{1, \ldots, F\}\},$$

where F is the number of optional features used across all subclassifiers in the hierarchical framework and $CNN(f)$ the convolutional module processing feature f.

The inference times of the CNN, LSTM, and fully connected modules are affected by the input and output size of the data passing through each module. Let us investigate the computations in each module individually.

For the CNN module, let $I = (H, W, C_{in})$ be an input image with size $H \times W$ pixels and C_{in} channels, and $K = (h, w)$ be the convolutional kernel of a CNN module with C_{out} channels. The kernel K is applied to each patch of the input map I, for each of the C_{in} input channels, for a total of HW patches. This leads to a total of $HW(C_{in}hw - 1)$ operations, and is repeated for each of the C_{out} channels, yielding a total time complexity of:

$$C_{out}HW(C_{in}hw - 1) \Rightarrow \mathcal{O}(C_{out}HWC_{in}hw).$$

Given the assumption that all optional features will be 1D, it becomes apparent that:

$$T_{CNN(wavelet)} \sim \mathcal{O}(C_{out}HWC_{in}hw) > \mathcal{O}(C_{out}WC_{in}w) \sim T_{CNN(f_{1D})},$$

as all optional convolutions are performed in parallel with the primary 2D convolution of the wavelet input features. Hence, we conclude that:

$$\max(T_{CNN(f)}) = T_{CNN(wavelet)},$$

which is common among all classifiers $C \in \{C_{flat}, C_{coarse}, C(a),\ a \in A \equiv \{walk, stairs, manual\}\}$. As such, we may only detect differences in the inference times of the classifiers $C \in \{C_{flat}, C_{coarse}, C(a),\ a \in A \equiv \{walk, stairs, manual\}\}$ by the remaining two modules, the LSTM cell and fully connected layer:

$$T_C \sim (T_{LSTM}, T_{FC}).$$

Let us now inspect the LSTM module's inference time. The time complexity of an LSTM module depends on the computations involved in its gates and the hidden state updates, given by the following set of equations at each timestep t:

$$\begin{aligned}
i_t &= \sigma\left(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}\right), \\
f_t &= \sigma\left(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}\right), \\
g_t &= \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}\right), \\
o_t &= \sigma\left(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}\right), \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t, \\
h_t &= o_t \odot \tanh\left(c_t\right),
\end{aligned}$$

where $x_t \in \mathbb{R}^d$ is the input to the LSTM module, $W_{im} \in \mathbb{R}^{h \times d}$ and $b_{im} \in \mathbb{R}^h$, $m \in \{i, f, g, o\}$, are the matrices and bias vectors representing the forward connections of the input, forget, cell and output gate, respectively, $W_{hm} \in \mathbb{R}^{h \times d}$ and $b_{hm} \in \mathbb{R}^h$, $m \in \{i, f, g, o\}$ are the matrices and bias vectors representing the recurrent connections of the above gates, and $h_{t-1} \in \mathbb{R}^h$ is the hidden state of the LSTM module. The computations for i_t, f_t, g_t and o_t are the same, so it is sufficient to investigate the time complexity of one of them, say the input gate i_t.

Inside the non-linearity, we have the following operations: $W_{ii}x_t$ matrix-vector multiplication requiring hd operations, $W_{hi}h_{t-1}$ matrix-vector multiplication with h^2 operations, and two vector additions, with h operations each. Assuming ReLU as the activation function, σ is applied element-wise, which therefore leads to an additional h operations, giving a total time complexity of $\mathcal{O}(hd + h + h^2 + h + h) = \mathcal{O}(h(d + h + 3)$. Repeating for the forget, cell and output gates, we get a total time complexity of $\mathcal{O}(4h(d + h + 3))$.

The LSTM cell state c_t and hidden state h_t involve dot product operations, which translate to element-wise multiplications, hence leading to h operations each. Consequently, we have a time complexity of $\mathcal{O}(2h)$ for c_t, as it involves two dot product operations, and also a time complexity of $\mathcal{O}(2h)$ for h_t, with h operations arising from the dot product and another h operations from applying elementwise the $tanh$ activation function. Hence, in total, we get a time complexity of:

$$T_{LSTM} \sim \mathcal{O}(4h(d+h+3)+2h+2h) = \mathcal{O}(4h(d+h+4)),$$

which is dependent on the dimensionality d of the input to the LSTM cell, x_t, and the number of hidden cells h.

To analyse the inference time of the LSTM cell within the contexts of C_{flat} and C_{hier}, recall that C_{flat} is defined to include all optional features present across the various subclassifiers of the hierarchical framework. Consequently, the LSTM cell in C_{flat} must process a significantly larger input space, as it incorporates the fusion of all optional features along with the output of the $CNN(wavelet)$ module. This expanded input space results in considerably longer inference times for C_{flat} compared to each of the subclassifiers in C_{hier}, where the input dimensionality is effectively reduced due to the modular handling of features. As a result, we expect:

$$T_{LSTM_{flat}} \geq \max(T_{LSTM_{coarse}}, T_{LSTM_{C(a)}}),\ a \in A \equiv \{walk, stairs, manual\}.$$

Finally, let us inspect the effect of the final fully connected layer on the models' inference times. Each subclassifier in C_{hier} operates on a smaller set of activity classes compared to C_{flat}. Let N be the number of output classes; the scope N_a of each $C_{coarse}, C(a),\ a \in A \equiv \{walk, stairs, manual\}$ is either i) a direct subset of the detailed activities in C_{flat}, or ii) a set of higher-level activities representing groups of activities from C_{flat}, i.e., it always holds that $N_a < N_{flat},\ a \in A \equiv \{walk, stairs, manual\}$. As a result, the different scopes of activity classes in $C(a),\ a \in A \equiv \{walk, stairs, manual\}$ lead to smaller matrices in the tensor multiplications of the final fully connected layers of the

subclassifiers of our hierarchical framework. With the mathematical formula of a fully connected layer being:

$$y = xW^T + b,$$

where $x^{\mathbb{R}^{1\times M}}$ is the input to the fully connected layer, $W^{\mathbb{R}^{N\times M}}$ is the weight tensor, and $b^{\mathbb{R}^{1\times N}}$ is the bias vector of the fully connected layer, it holds that the time complexity of this operation T_{FC} depends on the shape of tensors W and x, requiring NM operations, i.e., $T_{FC} = \mathcal{O}(NM)$. Hence, as $N_{flat} > N_a$ for any $a \in A \equiv \{walk, stairs, manual\}$, we conclude that:

$$T_{FC_{flat}} > T_{FC_{coarse}} \; and$$
$$T_{FC_{flat}} > T_{FC_{C(a)}} \; \forall \; a \in A \equiv \{walk, stairs, manual\}.$$

Conclusions on inference times To summarise, we have established that:

- The inference time T_{hier} of the hierarchical framework is primarily affected by the inference time required to process the high-level activity information, i.e., T_{coarse} and only in part by lower-level subclassifiers, based on their frequency of activation:

$$T_{hier} = T_{coarse} + \sum_a p_a N_R T_{C(a)}, \; a \in A \equiv \{walk, \; stairs, \; manual\}.$$

- The inference times for all classifiers C_{flat}, C_{coarse} and $C(a)$, $a \in A \equiv \{walk, stairs, manual\}$ are affected by the time required to parse the LSTM and fully connected modules, for which it holds:

$$T_{LSTM_{flat}} \geq \max(T_{LSTM_{coarse}}, T_{LSTM_{C(a)}}), \; a \in A \equiv \{walk, stairs, manual\},$$

$$T_{FC_{flat}} > T_{FC_{coarse}},$$
$$T_{FC_{flat}} > T_{FC_{C(a)}} \; \forall \; a \in A \equiv \{walk, stairs, manual\}.$$

Combining the above insights, we conclude that by designing the hierarchical framework such that the high-level classifier C_{coarse} relies minimally on optional hand-engineered features, it is possible to achieve reduced inference times compared

to the flat classification approach. This reduction depends on the frequency with which lower-level classifiers are activated; as not all activities require classification at the highest level of granularity at all times, the hierarchical framework enables more efficient inference by selectively engaging detailed classifiers only when necessary. This selective activation allows the inference time within the hierarchical framework to remain more contained than the flat classification approach.

In the remainder of this section, we will experimentally show the inference time capabilities of our proposed hierarchical framework.

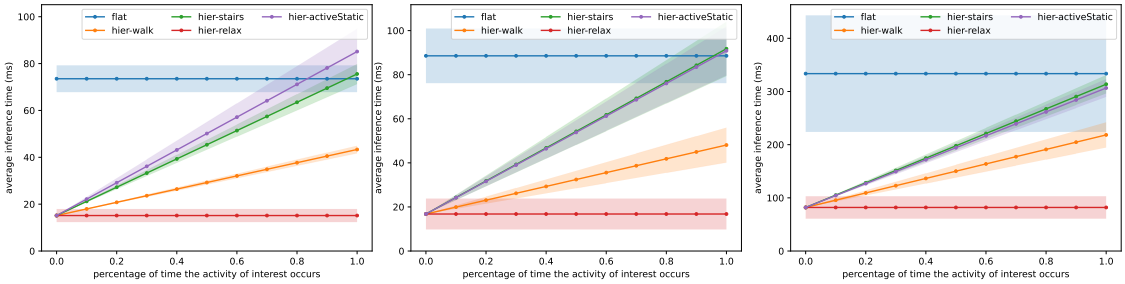
Inference time evaluation on server, tablet, phone To examine the computational efficiency of our hierarchical framework, we deploy C_{flat} and C_{hier} in three devices of varying computing power: a workstation server and two mobile devices (a Windows surface tablet and an Android phone). Table 5.4 summarises the hardware configuration for these devices.

Table 5.4: Hardware configuration for devices testing inference times for C_{flat} and C_{hier}.

Device	Model	Specifications	
workstation server	Linux server	OS:	Ubuntu 16.04.7 LTS
		CPU:	56-Core { Intel® Xeon® Gold 5120 2.20GHz
		GPU:	GeForce RTX 3070
tablet	Microsoft Surface Pro 8	OS:	Windows 11 Home edition
		CPU:	Quad-core { 11th Gen Intel® Core™ i5-1135G7 Processor
		GPU:	Intel® Iris® Xe
android phone	OnePlus 7T	OS:	Android 10, OxygenOS 12.1
		CPU:	Octa-core { 1 x 2.96 GHz Kryo 485 3 x 2.42 GHz Kryo 485 4 x 1.78 GHz Kryo 485
		GPU:	Adreno 640 (700 MHz)

Fig. 5.8 summarises the average inference times for the two models C_{flat} and C_{hier} in the three examined devices. More specifically, we calculate the average time required for each model to effectively respond to queries regarding which activity is performed (*exploration*) or whether a particular activity is performed

(*awareness*). As discussed in Section 5.5.1.3, C_{flat} and C_{coarse} need to be activated for all *exploratory* and *awareness* requests, denoted N_R. However, each detailed subclassifier $C(a),\ a \in A \equiv \{walk, stairs, manual\}$ in the hierarchical framework will only be activated for the subset of requests $N_{R(a)}$ that require information about an activity at a finer granularity, i.e., with a probability $p(a) = \frac{N_{R(a)}}{N_R} \in [0, 1]$. In this experiment, we measure the inference times of C_{flat} and C_{hier} by varying the probability of activation of each subclassifier for an exhaustive set of probabilities, where $p(0)$ denotes the detailed information is never required, and $p(1)$ denoting the detailed information is required at all times.



(a) workstation **(b)** tablet **(c)** phone

Figure 5.8: Average inference times of C_{flat}, C_{hier} for each HT relative to the percentage of time the HT is performed.

Our proposed approach reduces the inference times to identify activities in all three testing devices for probabilities of activity occurrences as high as 0.85 for the worst-case scenario. In practice, however, it is unlikely an activity (or set of activities) will be performed at such a high frequency during the day.

The experiment described above illustrates the ability of C_{hier} to effectively (and often significantly) reduce the computational resources required to identify each activity individually, compared to C_{flat}. However, it cannot demonstrate C_{hier}'s inference times capabilities in the real world, where the user interchanges a multitude of activities during the day at different frequencies.

To illustrate the effectiveness of C_{hier} in this realistic set-up, we conduct two practical case studies: 1) an assisted living scenario for elderly people [383] and 2) a development monitoring application for children with ADHD [384, 385]. In each scenario, we evaluate the computational efficiency of our framework based on the

average CPU processing time required for model inferences if we run the mobile application continuously for one day.

As smartwatch inertial data for these scenarios are unavailable, we rely on behavioural studies presented in [383, 385], which detail common activity patterns among older adults and children with ADHD. These patterns are mapped onto the activities defined in our hierarchical framework. For instance, the activities "read" and "watch TV", categorised under "Leisure" in the referenced studies, are assigned to our "relax" class. For activities that could be assigned to multiple classes in our hierarchy, we assume that the time spent on each activity is evenly distributed across the relevant classes. For example, 114 minutes spent on the activity "play games" are divided equally, with 57 minutes allocated to the "relax" class and 57 minutes to the "walk" class.

Following this mapping process, we normalise the time allocated to each class of our framework over the course of a day, yielding the probability of occurrence for each class and, consequently, the likelihood of activating the corresponding subclassifier. Using these probabilities of activation, along with the experimentally established average inference times for each subclassifier in our framework, as depicted in Fig. 5.8, we report the expected inference times under the real-world scenarios that reflect the activity patterns typical of older adults and children with ADHD, should each subclassifier be activated with the estimated frequency.

It is important to note that this study is purely numerical and relies on a few key assumptions. Due to the unavailability of relevant data for these case studies, we hypothesise the following scenario: older adults and children with ADHD engage in the activities reported in the referenced studies while wearing a smartwatch capable of recording inertial data. The simulated scenario assumes that the collected data could be processed by C_{flat} and C_{hier}, leading to: i) continuous activation of C_{flat}, ii) continuous activation of C_{coarse} within the hierarchical framework, and iii) activation of each subclassifier within C_{hier} with a probability equal to the frequency of the corresponding activity, as reported in the studies and mapped to our Granular Activity Tree. Note that this analysis further assumes that C_{coarse}

operates with near-perfect accuracy in identifying activities at a high level. While this assumption is idealised, C_{coarse} has demonstrated strong performance with an F1-score of 92.41% (as shown in Table 5.11). Therefore, we consider this assumption unlikely to significantly impact the observed outcomes, as the investigation focuses on inference times rather than classification accuracy.

We consider two modes in each scenario: First, we account for the time spent sleeping during the night as part of the daily activities performed. This mode is supported by the advancements in the battery life of many modern smartwatches (e.g., Empatica's Embrace2, Fitbit's Sense, etc.), which allows for consecutive use of the smartwatch for an estimated 2-6 days before charging is required, as reported in their corresponding sites. The increase in battery life could significantly aid the monitoring of fatal events, particularly for the elderly [386–388], and is thus of utmost importance to consider. Second, we will look into the more realistic scenario, where the smartwatch will regularly need to be charged, a task most likely performed while the user is asleep.

Assisted living for elderly people In recent years, the advances in medicine and the increase in life expectancy have formed the phenomenon of the ageing population; people typically live until old age but are often faced with age-related diseases (e.g., dementia, Parkinson's) that, as progress, require regular monitoring to ensure the patient's well-being; other times, they may develop walking difficulties and instability, often making them prone to falls [388].

Based on the above, it becomes evident that older adults could benefit from constant monitoring to be assisted appropriately when required. Under the scope of our work on HAR for ADL in this work, we investigate how our hierarchical framework can better help remote monitoring of the elderly by inferring their regular ADL in reduced time compared to C_{flat}.

To calculate the probability distribution of the activities typically performed by older adults under the scope of C_{hier}, we cast their daily patterns [383] to our corresponding HT. The breakdown and mapping between the two sets of

activities and the calculated probability distribution are depicted in Table 5.5 and Table 5.6, respectively.

Table 5.5: Estimating probabilities of activities for elderly people's assistive living.

Category	Time (min/day)	Detailed Activities	High-level Activities	Time (min/day)
Leisure	457	exercise	walk	114
		play games	relax	57
			manual task	57
		read	relax	114
		watch TV	relax	115
Work	41	type	manual task	21
		write	manual task	20
Domestic Work	211	shop	walk	42
		home/car maintenance	walk	21
			manual task	21
		childcare	walk	14
			relax	14
			manual task	14
		clean	walk	21
			manual task	21
		cook	manual task	43
Sleep	570	sleep/nap	relax	570

Table 5.6: Mapping probabilities of daily activities to high-level watchHAR labels for assistive living of older adults.

High-level Activities	Probability of activity (excl. sleep)	Probability of activity (incl. sleep)
walk	26.53%	16.58%
relax	48.82%	68.02%
manual tasks	24.65%	15.40%

The results are summarised in Table 5.7. Inference times are significantly improved with C_{hier} both for including and excluding the sleep mode.

Table 5.7: Inference times in ms for elderly people monitoring case study of ADL distribution.

Device	C_{flat}	$C_{hier,excl.sleep}$	$C_{hier,incl.sleep}$
workstation	73.53 ± 5.73	**34.69 ± 4.41**	**27.36 ± 3.81**
tablet	88.56 ± 12.40	**38.24 ± 8.59**	**30.19 ± 8.00**
phone	333.60 ± 109.65	**139.14 ± 20.27**	**117.68 ± 20.59**

Development monitoring for children with ADHD In younger ages, excessive usage of digital devices (video games, social media, etc.) has been linked to the increase of diseases such as diabetes, obesity, etc. [384]. At the same time, many movement disorders, such as ADHD, often become visible from a young age [385]. Under this spectrum, monitoring typical everyday patterns for children becomes a valuable tool for identifying symptoms or tracking the progress of those conditions. Children with ADHD, in particular, have significantly benefited from a well-defined daily routine [385], which our approach to ADL can assist in monitoring.

Similarly to the elderly case described above, we infer the probability distributions of the activities performed based on the proposed daily routine for children with ADHD [385]. The breakdown and mapping between the two sets of activities and the calculated probability distribution are depicted in Table 5.8 and Table 5.9, respectively.

The results are summarised in Table 5.10. Inference times are significantly improved with C_{hier} for the including-sleep mode. For the no-sleep mode, we still identify considerable speedup in the inference process, despite the probability distribution for the leaning towards the "manual" activities, which are more computationally demanding, as shown in Fig. 5.8.

5.5.2 Feature engineering under hierarchical framework

In the previous section, we demonstrated how our modular classification algorithm and operates during training and inference. In this section, we will present in detail how our suggested feature engineering module fits under the overall hierarchical framework.

Table 5.8: Estimating probabilities of activities for ADHD development monitoring.

Category	Time (min/day)	Detailed Activities	High-level Activities	Time (min/day)
Morning Routine	60	go to bathroom	walk	2
		wash face	manual task	8
		comb hair	manual task	5
		get dressed	manual task	5
		eat breakfast	manual task	25
		brush teeth	manual task	10
		leave house	manual task	5
School	420	type	manual task	105
		write	manual task	105
		read book	relax manual task	53 52
		exercise	walk	105
After-school routine	30	snack	manual task	15
		unwind	relax	15
Homework	60	type	manual task	17
		write	manual task	19
		read	manual task	19
		stretch, bathroom break	walk relax manual task	1 2 2
Leisure	90	play	walk manual task	22 23
		relax	manual task	45
Dinner	105	prepare table	walk manual task	25 5
		wait for food	relax	15
		eat, drink	manual task	30
		clean table	walk manual task	25 5
Bedtime routine	60	bath, teeth	manual task	30
		read, chat	relax	30
Sleep	600	sleep	relax	600

Table 5.9: Mapping probabilities of daily activities to high-level watchHAR labels for children with ADHD.

High-level Activities	Probability of activity (excl. sleep)	Probability of activity (incl. sleep)
walk	21.82%	12.63%
relax	13.94%	50.18%
manual tasks	64.24%	37.19%

Table 5.10: Inference times in ms for real-life case studies of ADL distribution.

Device	C_{flat}	$C_{hier,excl.sleep}$	$C_{hier,incl.sleep}$
workstation	73.53 ± 5.73	**55.34 ± 7.17**	**38.41 ± 5.34**
tablet	88.56 ± 12.40	**59.60 ± 10.08**	**41.57 ± 39.92**
phone	333.60 ± 109.65	**191.73 ± 18.52**	**145.49 ± 19.62**

5.5.2.1 Hand-engineered feature extraction

To evaluate our proposed modular approach to hand-engineering features, we will compare the classification accuracy of the $C(coarse_{HAR}, GL_1)$ and $C(walk, GL_2)$ classifiers with and without the use of input gravity features. As summarised in Table 5.11, the gravity features do not assist the high-level classification task but are essential for accurate identification in the lower-level $C(walk, GL_2)$. As such, we demonstrate that we can effectively extract hand-engineered features on demand, only when required by the inference task.

Table 5.11: HAR accuracy in modular feature engineering.

Accuracy Metric	C_{coarse_{HAR},GL_1}		C_{walk,GL_2}	
	with grav	w/o grav	with grav	w/o grav
Precision	91.78%	**93.36%**	**87.22%**	78.47%
Recall	**93.05%**	91.58%	**89.39%**	76.88%
F1-score	92.41%	**92.46%**	**88.29%**	77.67%

5.5.2.2 Wavelet feature extraction

For the prototype 2-layer hierarchical model defined above, we used the Biorthogonal family of wavelets (Bior2.2) [213] to calculate the 2D scalograms across both levels

of the hierarchy. The varying effect of the gravity feature across the two granularity levels, however, hints at the possibility of also tailoring the selection of mother wavelet to each level of the hierarchy. Here, we will evaluate the average f1-score performance across five candidate mother wavelets, for all the classifiers involved in the hierarchical framework. As depicted in Table 5.12, the choice of mother wavelet can significantly affect the classification performance. There is also no unique wavelet for the HAR task, rather a separate wavelet is optimal for each classifier in the hierarchical framework. Thus, the mother wavelet is an important hyperparameter to be tuned. Table 5.13 summarises the effect of the mother wavelet choice on the performance of C_{hier} for the choice of the overall best-performing wavelet `sym5` and the optimal combination of wavelets across the classifiers that constitute C_{hier}.

Table 5.12: Effect of wavelet choice on average F1-score performance.

Mother Wavelet	**average f1-score**				
	C_{coarse_{HAR},GL_1}	C_{walk,GL_2}	C_{stairs,GL_2}	C_{manual,GL_2}	C_{hier}
`db3`	92.86%	**88.52%**	**84.68%**	71.35%	80.95%
`sym5`	92.55%	87.77%	79.04%	**73.36%**	**80.98%**
`coif6`	90.62%	77.14%	73.11%	69.27%	72.53%
`rbio2.4`	93.09%	88.19%	80.42%	71.39%	80.19%
`bior2.2`	**93.10%**	87.31%	79.43%	72.44%	76.67%
`optimal wavelet combination`	-	-	-	-	**81.97%**

Table 5.13: Effect of wavelet choice on average HAR accuracy.

Accuracy metric	$C_{hier,bior2.2}$	$C_{hier,sym5}$	$C_{hier,comb^*}$
average precision	76.12%	80.65%	**81.45%**
average recall	77.22%	81.31%	**82.50%**
average F1-score	76.67%	80.98%	**81.97%**

5.6 Related work

HAR based on IMU data has been an active area of research for almost two decades [74, 75, 201–204, 214, 215, 219]. While various sensor modalities, such as

cameras, object and ambient sensors, inertial sensors and physiological sensors [32–34, 177, 179, 180, 182], and approaches varying from traditional machine learning to deep models exist, **CHiHAR** focuses on leveraging privacy-preserving, light-weight solutions that are accurate and computationally efficient. This section examines prior research, highlighting their contributions to the field and identifying persisting challenges.

Non privacy-preserving approaches A variety of different sensors have been used to recognise human activities. Computer vision systems utilise various camera setups, including single cameras, multi-camera systems, stereo vision, infrared, and thermal imaging [39, 183, 184], as well as depth cameras [185–188]. Depth cameras in particular allow for approaches involving depth silhouettes and skeleton generation, however, challenges persist in ensuring data accuracy and real-time responsiveness, particularly in terms of self-occlusion issues and degraded silhouette quality [186–188]. In [189], a home activity summary system addressed unbalanced datasets and interactions involving multiple individuals. Similarly, [190] proposed a probabilistic framework for scene-description-based activity inference, validated on real-world videos, yet faced challenges related to dynamic, cluttered scenes. **CHiHAR** navigates the challenges of human privacy and cluttered spaces by utilising a single smartwatch as the sensing modality, carried by the user to be monitored without interferences from other occupants of the space.

Privacy-preserving approaches Privacy concerns with camera data have further driven interest in point-cloud sensors. For instance, RadHAR [35] and mActivity [36] explore mmWave radar as a privacy-preserving solution, though such systems often struggle with reduced spatial resolution compared to cameras.

Interaction-based HAR utilises ambient sensors to monitor and automate smart home environments. Early projects like GatorTech [191] and AwareHome [192] created proof-of-concept smart homes to study user interaction. However, these systems often require significant infrastructure, limiting scalability.

RFID technology has been widely employed [193–195], but its use introduces additional burdens such as electromagnetic exposure and high maintenance, prompting a transition to low-power, wireless systems. Wireless sensor networks (WSNs) have shown promise, with one study deploying 14 sensors to classify seven activities using HMM, CRF, and Naive Bayes classifiers [196–198]. While effective, these setups face challenges in energy consumption, placement optimisation, and latency.

Recent studies integrate hybrid models for enhanced performance. For example, [199] combines artificial neural networks (ANNs), support vector machines (SVMs), and HMMs, achieving notable improvements. Similarly, [200] demonstrates a 13.82% accuracy boost by incorporating SVM-based kernel fusion for action prediction. However, achieving real-time adaptability and robustness in diverse environments remains challenging.

IMU-based systems IMU-based HAR has evolved over two decades as the primary privacy-preserving approach, initially employing traditional machine learning models like SVMs and random forests [201–203]. Advances in deep learning have shifted focus to automatic feature extraction using convolutional (CNN) and recurrent neural networks (RNNs/LSTMs) [74, 75, 204]. Despite this progress, ensuring model generalisability and robustness across diverse users and activities remains a key issue.

Sensor fusion has also been explored extensively. Approaches range from early fusion, where input data from multiple sensors are combined, to shared-filter fusion, which models individual axes separately [214–216]. For instance, [219] achieved high performance with a convolutional network trained on foot-attached IMU data, generalising across users through adversarial augmentation. Most of these works, however, focus on a combination of sensors attached/carried at various body parts, some of which are unnatural for everyday use. While smartphone-based HAR has been extensively studied, smartwatches, despite their ergonomic placement, remain underexplored [66, 67]. Research focusing on wrist-attached sensors is promising

for recognising dexterous activities, though handling complex tasks and minimising hardware constraints are ongoing challenges.

Feature engineering in HAR systems More recently, deep approaches have seen a surge in research activity to bypass the need to design features manually: notably, [204] used deep neural networks for feature extraction before training on standard classification models. Similarly, [74, 75] built on this idea, suggesting a combined approach between hand-engineered feature extraction and deep models. In particular, they encoded the information from raw data in a visual representation, e.g., a spectrogram, which was then used in a deep convolutional neural network to perform activity recognition, similarly to our scalogram use on CNN-LSTM-based architectures.

Wavelet transforms have also shown their effectiveness in recognising human activity patterns from various input signals. Examples of input signals include video [205–208] and inertial measurement data [209–212]. In particular, discrete, complex, and continuous wavelet transform with the Symlet (Symlet 4) [205], Daubechies [206, 207], and Gabor [208] wavelets are employed to identify human activity from video frames accurately. Similarly, Biorthogonal (Bior2.2) [213], Haar [209], Mexican hat [212] and Daubechies (db10) [210] wavelets are used to recognise activities from inertial signals. However, the choice of wavelet and its effect on classification performance remains an open problem in these works.

Hierarchical HAR systems Hierarchical models in activity recognition offer an alternative to traditional flat classification models by leveraging the inherent similarities between activities. Rather than treating HAR as a single-stage multi-class classification problem, hierarchical approaches decompose it into a series of subclassifications, refining decision boundaries at each level. Prior research has explored both bottom-up and top-down hierarchical structures, each presenting unique advantages and limitations.

Bottom-up approaches typically focus on recognising finer-grained activity components before inferring higher-level activities. For instance, [67] proposed

a model in which human gestures were initially classified at individual sensor nodes, followed by a centralised inference step that combined these outputs with additional sensor readings to determine the specific activity. Similarly, methods such as hierarchical hidden Markov models (HHMMs) have been explored for HAR, where latent variables, representing subactions, are automatically detected and optimised within the model, leading to improved recognition performance compared to standard hidden Markov models [221]. In [223], the authors proposed a hierarchical method for the more complex problem of trajectory-based HAR based on hierarchical dirichlet process hidden markov models. Hierarchical breakdown and feature selection approaches for various walking activities were also proposed in [40, 41, 66, 67], focusing, however, only on the active/static separation or targeting the feature extraction to traditional machine learning classification, with, e.g., SVM or Naive Bayes classifiers.

On the other hand, top-down hierarchical approaches infer abstract activity categories first before classifying specific actions. This strategy enhances discrimination between similar activities without making explicit assumptions about temporal dependencies. For example, [228] proposed a two-stage classification system that initially categorises activities into broad states (static, dynamic, or transitional) before refining the classification using linear discriminant analysis and artificial neural networks. Camera-based hierarchical HAR solutions were proposed in [389, 390] to break down visual streams of activities to lower-level spatiotemporal components. These are promising directions; however, they assume the activity is known (or can be identified in the video frame) at each level of the hierarchy. They then decide on the sequence of activities that constitute a complex task. Other studies have leveraged deep learning models within hierarchical frameworks. In [229], the authors developed a two-stage convolutional neural network (CNN) that first optimised a CNN to classify abstract activity groups, followed by two separate CNNs to recognise specific activities. Our proposed approach **CHiHAR** also follows this directive, utilising a series of "nested" classifiers over the hierarchy.

Defining the hierarchy of activities In the aforementioned methodologies, the class hierarchy has been manually designed, which is also the case for our proposed approach **CHiHAR**. However, automated hierarchy generation is an active area of research, usually coupled with hierarchical activity modelling. Recently, a deep encoder-decoder approach was proposed to automatically learn low-level activities from high-level ones across two levels of hierarchy [224], which is a promising approach to assist automation of the GATs considered in this chapter. Ontological approaches also exist, particularly concerning bottom-up hierarchies that define higher-level activities based on the occurrences of simpler tasks (e.g., "preparing tea" is built over "reach cup", "use teabag", "boil water") [222, 231, 232, 234]. While some works also exist in the field of ontology regarding more contextual hierarchies [237–239], as is the case for the semantic grouping of activities over the granularity levels in **CHiHAR**, automated hierarchy generation is beyond the scope of this chapter.

CHiHAR Our framework inherits the qualities of hierarchy and wavelet transform that have proved effective for deep architectures [40, 74, 75, 212] and further extends the hierarchical properties to the feature engineering module. It operates effectively with IMU data from a single smartwatch, which has rarely been used as the sole sensor modality for HAR. Yet, it accurately identifies ADL at a much reduced time compared to flat classification, without augmenting the number of sensors to be worn by the user.

5.7 Conclusions

In this chapter, we proposed a hierarchical approach to learning ADL from a single smartwatch's IMU recordings based on time-frequency analysis using the wavelet transform. We demonstrated the model's inference time capabilities for the daily distribution of activities across two realistic scenarios; remote assistance for the elderly and children monitoring. We have also highlighted that feature engineering is an integral part of the hierarchical approach, significantly improving the model's performance when incorporated into the hierarchical scope. This paves the way for

the investigation of methods that optimise and automate feature extraction in each level of the hierarchy to maximise the end-to-end performance of the model.

Overall, our proposed framework effectively recognises activities at various granularity levels and minimises the computational resources required compared to standard non-hierarchical classification. We have demonstrated that structuring activities hierarchically opens up interesting directions for comprehending semantic concepts and establishing logical links within well-defined sets of activities.

However, the freestyle nature of human activities means that uncovering semantic similarities among groups of activities within a dataset may not suffice to capture the underlying structural intricacies in the semantics of activities within a new dataset. For example, while one person might define "walking" as the well-defined activity of continuously moving at an average pace, others may also account as walking the act of "taking a walk", which may include stopping to browse, engaging with other people, or grabbing a snack along the way. Nevertheless, shedding light on the variability in human conceptual understanding is essential for designing models that are generalisable and explainable in the real-world.

In this direction, we explore in the following chapter how to leverage a learned hierarchy to reveal structural patterns of incoming activities by mapping them to known concepts embedded in the hierarchy. This is a valuable approach towards understanding human perception of notions under different scopes and opens up intriguing avenues towards robust generalisation of models in their deployment in the real-world.

Chapter 6

Revealing semantic mappings across HAR datasets

6.1 Introduction

In the previous chapter, we proposed a centralised, hierarchical solution to human activity recognition that is capable of supporting the realistic concurrent demands of various mobile applications that co-exist in smart devices, such as smartwatches. In this chapter, we will explore the challenges in real-world HAR model deployment that may arise from key differences between its training set and incoming data. We propose a simple yet effective solution based on an inverse-hierarchy, to identify explainable mappings across independently collected datasets of similar semantic content and provide valuable information regarding the perception of activities across different users or datasets.

6.1.1 Motivation

The substantial growth and widespread availability of smart and wearable devices in the last few years, such as smartwatches, ear wearables, etc., has revealed a unique opportunity to collect extensive and continuous amounts of activity data [391]. Historically, datasets have been primarily collected and annotated in laboratory

conditions over carefully designed and controlled experiments. Meticulously curated, well-annotated datasets are essential for training models that can capture precise patterns; however, training deep learning models, in particular, requires large amounts of labelled data that are laborious to acquire [392]. Nowadays, wearables allow for the in-the-wild collection of data reflecting varying, naturalistic and often unexpected behaviours of the bearer of the devices. While gaining insight into the realistic behaviour of users can prove beneficial in many application domains, e.g., personalising workouts, smart homes and healthcare [367, 368], crowdsourced data can often be challenging to handle due to unscripted user behaviours, missing data and incomplete, noisy or incorrect annotations [393]. Given that deep neural networks are trained to be robust under the closed-world assumption, i.e., the incoming data at testing time are expected to be of the same distribution as the dataset used to train the model [246], the vast amount of varying activity data pushes the boundaries of building accurate new models and challenges the generalisation capabilities of existing models.

While significant research exists on out-of-distribution detection and active learning to identify and manage novel data at inference time, to our knowledge, no methodologies currently conceptualise the differences between training and inference data in a descriptive and explainable manner accessible to non-experts.

This thesis explores the explainability of incoming data by leveraging information embedded within a trained model. For the scope of this chapter, explainability refers to the ability to provide a semantic understanding of new data based on their conceptual similarities to known data from a source domain. This has two key benefits:

- From the perspective of a HAR system designer, it is not only useful to identify and handle entirely novel or significantly different activities, an area extensively studied in out-of-distribution detection and active learning, but also to recognise activities that are similar yet not identical to the training data. While some work towards this direction exists in the subfield of near-distribution detection, conceptually understanding how such activities deviate

from the model's existing knowledge can inform strategies for adapting and improving recognition models.

- From a user's perspective, such as an individual interacting with an activity recognition system via a wearable device, it is valuable to examine how the system's predictions compare with their own perception of performed activities. As discussed later in this chapter, such information can support both learning new skills correctly and monitoring changes in established behavioural patterns.

Hence, it is necessary to investigate ways that can mitigate such challenges and reveal the variability of incoming data characteristics with respect to the training dataset.

6.1.2 Challenges in real-world model deployment

In this work, we identify four key challenges arising from differences between a model's training set (represented as a "controlled/source dataset") and incoming data (represented as a "crowdsourced/target dataset") that may inhibit the deployment of a model in the real-world. These are summarised below and illustrated in Fig. 6.1:

1. *label content mismatch*, i.e., the set of classes between the two datasets may not be identical. While a subset of common semantics may exist across the two datasets, they are not guaranteed to contain the same set of semantics.

2. *timescale mismatch*, i.e., the labels may be assigned over different lengths of recordings across the two datasets. For example, samples in one dataset may be annotated over 3-sec windows, while in another, over 30-sec windows.

3. *label definition mismatch*, i.e., the annotators may perceive samples with identical labels across the two datasets differently. For instance, a user may consider themselves to be "walking" if they are continuously performing this activity over a fixed period, while another user may consider the same even if their walking was interrupted by intermittent resting periods

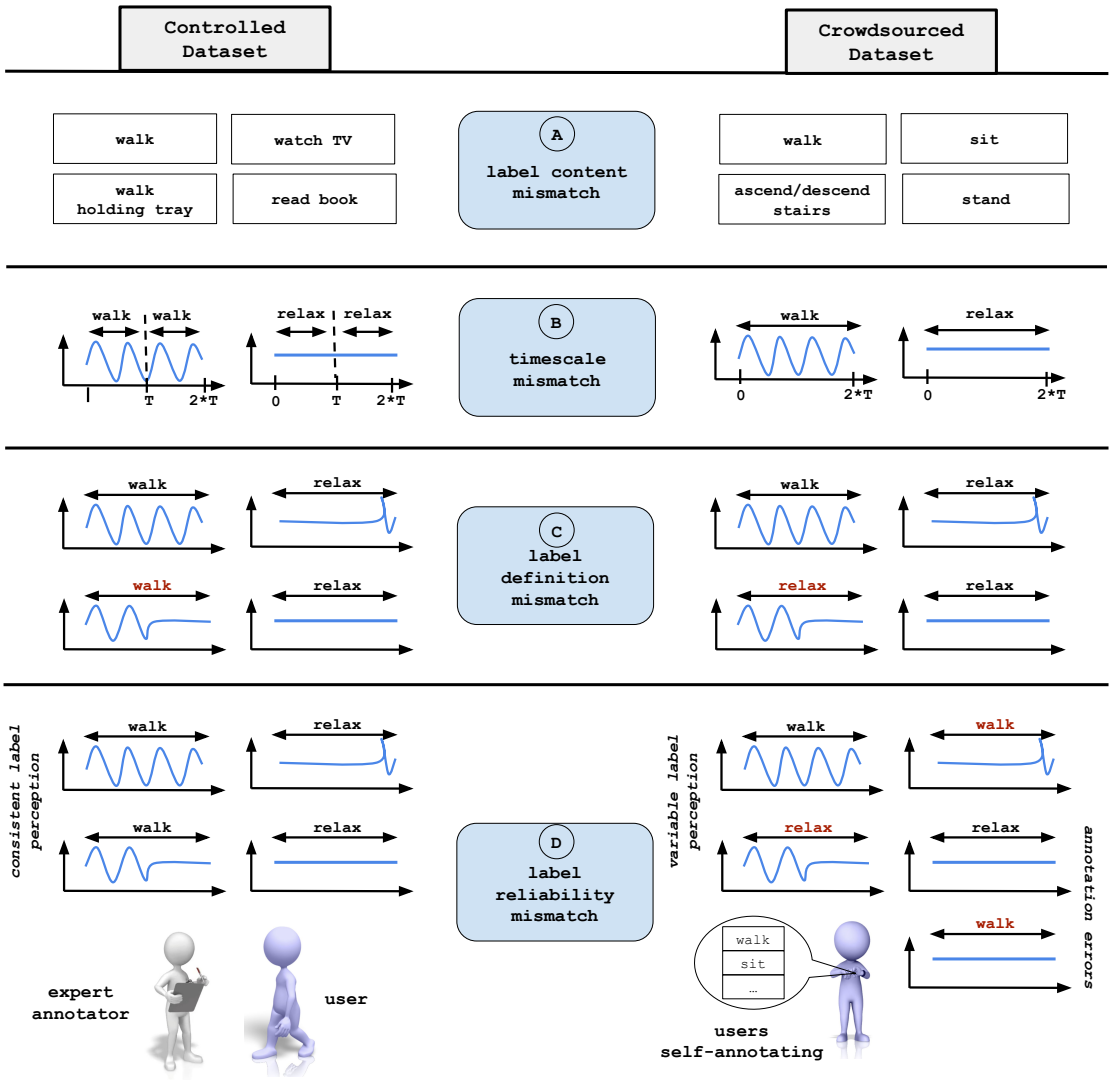


Figure 6.1: Challenges in semantic alignment of controlled and crowdsourced datasets.

(A) The set of class labels in the controlled and crowdsourced dataset may be semantically similar (e.g., watch TV / sit) but not identical. (B) Annotations may be assigned over windows of different sizes in the two datasets. For example, while in one dataset, we might have annotations every T sec, the frequency of annotations in the other dataset may be different, e.g., every $T' = 2T$ sec. (C) The underlying rules governing a class label may differ across the two datasets. For example, as demonstrated in the highlighted samples, the same recording may be assigned to a different class in each dataset, as the minimum threshold of the activity being performed may vary. (D) Particularly in real-world datasets, it is possible to have annotation errors or inconsistencies due to different user perceptions of each class label. For example, observing the highlighted samples: a sample with a significantly low percentage of walking patterns is assigned the label "walk", while a more active sample is annotated as "relax", i.e., no consistent underlying rule for label assignment within the crowdsourced dataset exists; at the same time, a static sample is erroneously annotated as "walk".

4. *label reliability mismatch*, i.e., we may not have insight into how accurate the labels of a target dataset are since they were collected without supervision. Such disparity may exist for different users within the target dataset, as each user interprets how each activity may be performed independently or because of erroneous label assignment during real-time annotation.

The *label content mismatch* is not a typical generalisation problem, where incoming data from a different distribution (e.g., different users or environments) need to be successfully identified by the model, or previously unseen data should be identified as such; however, it is related to the "out-of-distribution" detection problem, which is a component of generalisation mechanisms. While standard OOD detection has seen effective solutions when incoming samples belong to unidentified classes [240], it has not been studied in depth for cases when incoming samples may include new concepts, that however exhibit logical similarities to known classes, as is the case for *label content mismatch*.

More recently, a subset of OOD research has notably delved into the concept of near-distribution (nearD) detection [264, 268]. While this space focuses on distinguishing samples that resemble the model's training dataset from completely unrelated data, it has not explored, to our knowledge, mechanisms to identify the shared patterns between ID and nearD data that may be valuable for better adaptability of a trained model in new domains. Such information could also improve user experience, by showcasing to the user how their performed activities differ from the characteristics embedded in a train model.

The area of transfer learning offers solutions for a model to adapt in such new domains, where classes may be defined differently, resulting in *label content mismatch*. While knowledge transfer from the source model is enabled, these techniques also lack explainability mechanisms of the new domain. At the same time, existing approaches have explored model adaptations in diverse environments [394, 395], accommodating variations in users, devices, or device placements [42]; nonetheless, there is, to our knowledge, a notable absence of work directly supporting predictions at different input sizes (e.g., due to *timescale mismatch*).

Acknowledging the potential noise in incoming data annotations during real-world model deployment, leading to *label reliability mismatch*, recent efforts in learning from noisy labels have emerged [300, 302, 314, 327, 392, 396]. Yet again, however, the estimation of noise type typically adopts a numerical perspective rather than providing a semantic, explainable understanding of incoming data through the model's lens, a viewpoint we consider valuable to assist in the usability of the vast amounts of crowdsourced data.

By investigating these directions, it becomes evident that while many of the aforementioned real-world challenges have attracted significant research interest, there is a consistent lack of explainable solutions to the problems of model generalisation and knowledge transfer to achieve further learning. At the same time, to our knowledge, not all levels of semantic challenges we have highlighted have been addressed simultaneously in existing research.

6.1.3 Proposed approach and key contributions

In this work, we approach a *different* generalisation problem that aims to resolve the above challenges to not only enhance a model's generalisation and knowledge transfer capabilities under potentially noisy input but also to provide semantic explanations on incoming data of a similar nature to the training dataset.

In this direction, we introduce **SeMEDA**, a Semantic Mismatch Estimation and Dataset Alignment approach. **SeMEDA** aims to better *understand* crowdsourced data by establishing a *mapping* between the incoming data and our model's viewpoint of the world (i.e., the ID training labels) and using that mapping to capture inconsistencies and similarities in data annotations across the datasets. The significance of such information is multifold. Understanding the underlying structure of data that leads to unreliable labels can offer valuable insights into the characteristics of the target domain, thereby assisting designers of HAR models in developing more robust and generalisable frameworks capable of capturing these patterns. Simultaneously, providing users with feedback on how their performed activities deviate from expected patterns can be beneficial across various application

domains. For instance, this can aid in fostering personal hygiene habits in young children or monitoring motor skills in older adults.

In the first scenario, consider a young child who believes they have successfully brushed their teeth. However, their wearable device may detect prolonged periods of idleness during the activity, perhaps due to the child holding the toothbrush in their mouth while distracted rather than engaging in the necessary vigorous, repetitive motion of teeth-brushing. Such insights could help both the child and their parents in establishing a proper dental hygiene routine.

In the second scenario, an older adult in the early stages of Parkinson's disease may perceive themselves as being idle, yet their wearable might start detecting extended periods of manual activity due to increased tremor levels. This deviation from the user's expected behaviour, as defined by the classifier embedded in their device, could provide valuable information for both the individual and their medical team, facilitating further investigation into the progression of their condition, and new patterns that should be taken into account, directing how the user's wearable should adapt to correctly identify activities in the user's new way of living with their disease. Utilising these learned concepts, models can better *adapt* to the crowdsourced dataset domain, cautioning the user when incoming data labels should not be trusted and, simultaneously, revealing the underlying structure of the data that causes their given labels not to be trusted.

In summary, we make the following contributions in this work:

- We identify the variability between controlled and crowdsourced datasets even when these contain similar semantic classes. We highlight the necessity to find a mapping across such datasets to semantically understand a real-world dataset that may be different to the dataset used to train a model in terms of its *label content*, *timescale*, *label definition* and *label reliability*.

- We introduce **SeMEDA**, a Semantic Mismatch Estimation and Dataset Alignment approach that sequentially resolves the identified dataset mismatches, by introducing label-based and timescale-based mappings that reveal

underlying data dynamics and annotation inconsistencies in the new dataset. Our proposed approach allows for improved generalisation capabilities in trained models, increased learning capacity of new models in label and time domain, and for trustworthy curation of datasets from incoming noisy data.

- We demonstrate **SeMEDA** using two datasets: the **watchHAR** dataset that we introduced in Chapter 3.4, which was collected in a laboratory environment under careful instructions to the participants of the experiment, and the **ExtraSensory** [393] dataset, which was collected in-the-wild and was self-annotated by the users during data collection without any input or supervision from the researchers creating the dataset.

The remainder of this chapter is organised as follows: Section 6.2 presents the architecture of our proposed method and illustrates how the different modules work together. Section 6.3.4 introduces the algorithmic details for each of the system's components and explains how this achieves better semantic alignment of the datasets in each step. Section 6.4.4.2 summarises important implementation details for the framework's evaluation, and Section 6.5.4.2 then concludes the evaluation. Finally, Section 6.7 discusses limitations and future directions.

6.2 System Architecture

This section provides a high-level description of our system architecture and its main algorithmic components.

Our proposed approach aims to identify a semantic alignment between a controlled (source) and a crowdsourced (target) dataset by identifying and mitigating the challenges that may occur due to differences in various levels of their semantic understanding while at the same time providing explainable representations of incoming crowdsourced data. Estimating and resolving such challenges is vital for a model trained on a controlled dataset to effectively generalise on a (or the subset of a) crowdsourced dataset with a semantically similar set of classes. It also aids knowledge transfer from the source domain to assist the training of models using

the crowdsourced dataset, particularly when the dataset's given annotations are inconsistent or unreliable. In our proposed framework, mitigating these challenges is achieved through four main components, depicted in Fig. 6.2 and outlined below:

1. **Estimation and mitigation of *label content mismatch***, to identify the subset of semantically similar classes across the two datasets and define a mapping between them. This module borrows from the *hierarchical approach to learning* we defined in Chapter 5.7 to establish a *label mapping* to a shared semantic space that encompasses all classes represented in the two datasets.

2. **Estimation and mitigation of *timescale mismatch***, to investigate and resolve how annotating samples at different window sizes affects the generalisation capabilities of models trained on the controlled dataset and the learning capacity of models trained on the crowdsourced dataset. A *timescale mapping* ensures predictions can effectively be made in the incoming target data timescale.

3. **Estimation and mitigation of *label definition mismatch***, to identify and reveal the different underlying rules governing the annotations of the two datasets. Unveiling these distinct rules establishes a *cross-domain mapping* across the datasets.

4. **Estimation and mitigation of *label reliability mismatch***, to identify and resolve the lack of consistent underlying rules governing the annotations and highlight misannotated instances in the crowdsourced dataset. A *target-domain mapping*, similar to the above-mentioned *cross-domain mapping*, is employed to reveal the full spectrum of activity task perceptions within the target dataset and highlight any annotation errors that may exist.

In summary, our framework operates as follows. First, we identify the set of dataset mismatches that are observable through the datasets' descriptions and initial investigation, for example, the set of classes in each dataset, the frequency of annotation, any specified rules regarding how the activities should be performed

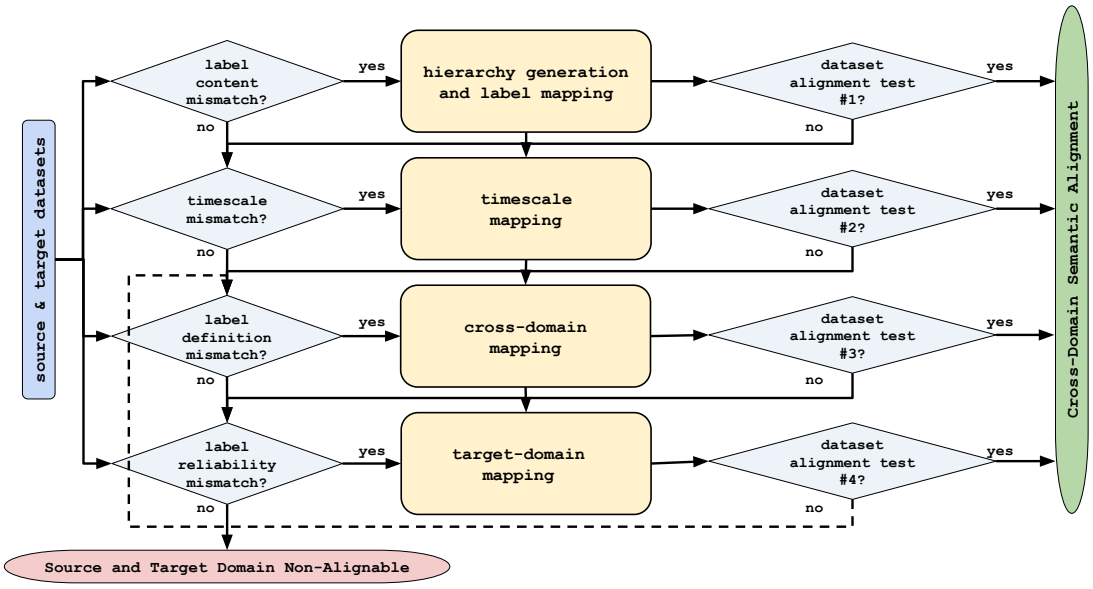


Figure 6.2: System Architecture.

Source & target datasets: Assume a source and a target dataset defined over a set of semantically similar but not identical activity classes (*label content mismatch*). The datasets may also be annotated at different frequencies (*timescale mismatch*), may be defined with different underlying rules to govern each activity class (*label definition mismatch*) and have different levels of annotation noise, observed as annotation inconsistencies or errors (*label reliability mismatch*).

Dataset alignment: If there exists a semantic space where the source model can generalise to the target dataset and a new target model can utilise knowledge from the source domain to distinguish among the target dataset's samples, the datasets are *semantically alignable*.

Estimation and mitigation of *label content mismatch*: If there exists a *label content mismatch*, estimate the shared semantic space as a level of semantic abstraction that encompasses all classes from both datasets. We denote this step as *hierarchy generation and label mapping*.

Estimation and mitigation of *timescale mismatch*: If datasets are not aligned, and there exists a *timescale mismatch*, establish a *timescale mapping* that allows for predictions of both source and target models in the target timescale.

Estimation and mitigation of *label definition mismatch*: If datasets are still not aligned, there might also exist a *label definition mismatch*. To resolve, attempt to identify the target dataset's underlying activity class patterns through a *cross-domain mapping*.

Estimation and mitigation of *label reliability mismatch*: If datasets remain non-aligned, there may further exist a *label reliability mismatch*. To resolve this, establish a *target-domain mapping*, to provide explainable representations of incoming data that may reveal inconsistencies or erroneous annotations within the target dataset.

Output: If dataset alignment is accomplished, the datasets are alignable in the shared semantic space under the employed mappings. Otherwise, the datasets are considered non-alignable.

and any known annotation noise levels. We then sequentially attempt to resolve the mismatches in the aforementioned order by establishing label-based and time-based mappings that aim to identify a shared semantic space where dataset alignment can be achieved. A mismatch is considered resolved under the selected mitigation mechanism if its corresponding *alignment test* is passed; instead, a failed alignment hints at the existence of additional mismatches to those originally identified. These tests inspect the models' generalisation and learning capabilities in the common mapped semantic space.

If a mitigation mechanism is employed to resolve a mismatch at a certain level, it is then also used by all further mitigation approaches in subsequent mismatches to be resolved. Once all mismatches are resolved, indicated by a positive alignment test, the semantic alignment of the two datasets has been achieved. This allows for both datasets to be used with trustworthiness for further downstream tasks in their shared semantic space.

Having presented an outline of our framework, in the following section, we will delve into the algorithmic details of our proposed methodology.

6.3 Algorithms

In the previous section, we provided a high-level description of our framework. In this section, we will discuss the details of our proposed approach. We will introduce the mismatch identification mechanisms and their corresponding mitigation methodologies for each module in our framework.

Our proposed methodology assumes a controlled (source) and a crowdsourced (target) dataset, each defined over a set of single-activity (i.e., non-composite) classes. In reality, the classes in the two datasets may be identical (semantically ID), logically similar (semantically nearD) or purely distinct (semantically OOD); in this work, however, we only consider the special case where a subset of class labels are identical across the two datasets (ID classes), and the remaining labels are semantically similar, albeit non-identical to the ID class labels (nearD classes),

as the ID and OOD have been explored in the research areas of transfer learning and out-of-distribution detection.

In this context, we aim to identify three key subsets of samples within the nearD crowdsourced dataset:

- a trustable ("pure") subset, that presents the behaviour indicated by the target sample's annotation throughout the largest part of the recording,
- an ambiguous ("polluted") subset, where the annotated activity labels presents in the recording, albeit interchanged with other activity concepts,
- an untrustable ("mislabelled") subset, for which the annotated activity does not appear (at all or significantly) within the sample; instead, other activities are predominantly identified throughout the recording's duration.

Our goal is to estimate and mitigate the variations in the semantic definitions of the two datasets by revealing a shared semantic space, and the trustworthy behaviours within it where: (i) a source model, trained on the controlled dataset, can be adapted to generalise on incoming samples from the target dataset; and, (ii) new models can be robustly trained on the target dataset, either directly or by utilising the knowledge encoded in the source model. The datasets are considered *alignable* if both properties hold. To verify these conditions, we employ *dataset alignment tests*, which measure the models' performances through their predictive accuracy.

We define the following key concepts, that are utilised in the remainder of this section:

- a *source recording* is a sample from the source (controlled) dataset,
- the *source-timescale* is the window size corresponding to the annotation frequency in the source (controlled) dataset,
- a *target recording* is a sample from the target dataset,
- the *target-timescale* is the window size corresponding to the annotation frequency in the target dataset.

6.3.1 Estimation and mitigation of *label content mismatch*

To identify the *label content mismatch* between a source and a target dataset, we need to compare their respective class labels. As discussed above, we study the special case where only identical and semantically similar class labels exist in the two datasets. While achieving a 1-1 mapping between the source and target labels is not possible due to the existence of nearD classes, such a semantic mapping may exist at a coarser level of semantic abstraction.

To establish this mapping at training time, we first inspect all the class labels that appear in the source and target dataset and identify distinct groups of classes across the two datasets that share logical semantic similarities and can be described by a coarser activity label. Then, we introduce an *inverse hierarchy* approach, inspired by our hierarchical HAR framework in Chapter 5.7. For each dataset, we generate an inverse-hierarchy that groups its relevant corresponding activities to the identified coarser activity labels. This results in two *parallel hierarchies*, for which a 1-1 *label mapping* can be established across the identified high-level activity representations.

Similarly to the design of Granular Activity Trees (GAT) in Section 5.3.2.2, the design of the inverse-hierarchical structure offers considerable flexibility, particularly in the selection of coarser activity labels. This design can be informed by the model designer's expertise or tailored to the specific application context. However, to ensure effective alignment, it is crucial that a one-to-one correspondence exists at some level of granularity between the source and target datasets, as the method aims to capture the contextual similarities among the two datasets. To design the inverse-hierarchy, we follow the same process as detailed for the hierarchy design in Section 5.3.2.2, with the only difference that at training time, we account for the detailed labels from both datasets to design the hierarchy. At inference time, the selected hierarchy is utilised to project concepts from the source dataset to the target dataset, based on the selected mapping.

If certain target classes cannot be mapped to conceptually similar classes in the source dataset, they should be highlighted during the hierarchy design as significantly different, for example by categorising them under an "other" class.

However, since this study specifically focuses on near-distribution samples, we do not anticipate such occurrences. Consequently, we proceed without incorporating an "other" class, though the methodology could be readily extended to accommodate this by integrating OOD detection mechanisms in the source models, that can capture such data that vary significantly in their semantic concepts from the training set, such as, for example, the work in [268].

For illustrative purposes, let us consider the sample datasets in Fig. 6.3. In this example, only the class `walk` is identical across the two datasets. However, the classes across the two datasets can easily be summarised under two coarser activity labels: `walk` and `relax`. We then build two parallel hierarchies: the

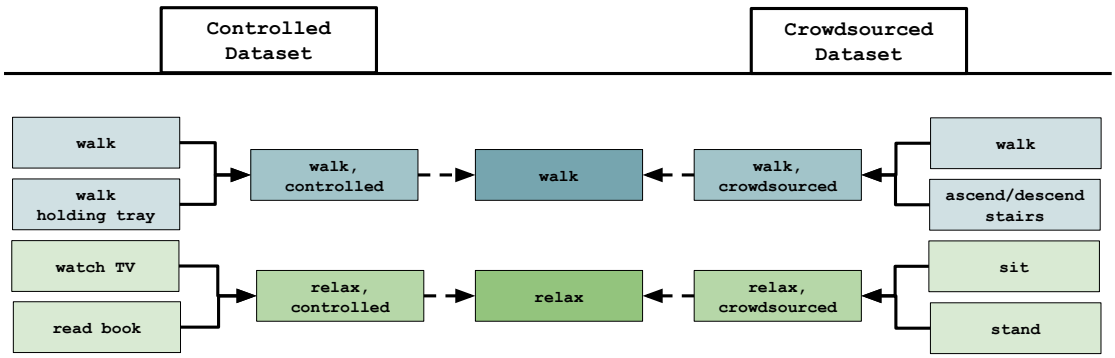


Figure 6.3: Estimation and mitigation of *label content mismatch* - an illustrative example.

Sample controlled and crowdsourced datasets suffering from *label content mismatch*, as only one class label is identical across the two datasets. However, the remaining class labels are *semantically similar*; hence, it is possible to group the activities in each dataset at a coarser level of semantic abstraction (namely, the tasks `walk` and `relax`). This shared semantic space encompasses all class labels across the two datasets, allowing for a 1-1 mapping between the source and target labels in this higher abstraction level in the semantic activity descriptions hierarchy.

classes `walk` and `walk holding tray` in the controlled dataset can be jointly seen as the more generic class $\texttt{walk}_{\texttt{controlled}}$, and the classes `watch TV` and `read book` can be accounted as relaxation time under the $\texttt{relax}_{\texttt{controlled}}$ label. Similarly, the classes `walk` and `ascend/descend stairs` can be summarised under the label $\texttt{walk}_{\texttt{crowdsourced}}$, and the classes `sit` and `stand` under the label $\texttt{relax}_{\texttt{crowdsourced}}$ in the target dataset. At this higher level of semantic abstraction, creating a 1-1

mapping across the corresponding grouped labels of the two datasets is possible, paving the way for seamless predictions on incoming samples from any of the two datasets in this shared semantic space.

We perform a dataset alignment test to identify whether resolving the *label content mismatch* suffices for dataset alignment. Suppose the source model generalises on the target dataset, and a target model learns robustly from the target dataset in the identified shared semantic space. Then, we consider the mapping effective, and source and target domains are aligned under the generated mapping. This includes cases where the datasets are defined at different timescales, but generalisation of the source model at the target-timescale may not be considered necessary. If this is not the case, or if there is a further requirement for the source model also to learn to generalise at the target-timescale, there is a *timescale mismatch* to be addressed.

In the following section, we will inspect how to establish predictions at a target-timescale, to resolve the *timescale mismatch*.

6.3.2 Estimation and mitigation of *timescale mismatch*

A *timescale mismatch* is identified when labels across a source and a target dataset are assigned over windows of different lengths. If the source model is required to learn to generalise on the target timescale, we need to establish a *timescale mapping*, to ensure that class predictions can be made at the target-timescale.

Assuming a *semantic hierarchy* and a *label mapping* are in place to resolve any existing *label content mismatch*, we employ the following process to address the *timescale mismatch*. We create windows of size source-timescale for each recording in the target dataset. We then evaluate each of these windows sequentially on the source model at the source-timescale, generating a sequence of predictions for each target recording. We enable a memory mechanism to hold the generated relevant information for the set of N consecutive windows that constitute a target recording. With this information at hand, we introduce two alternative approaches to generalising at the target domain, also summarised in Fig. 6.4 through an illustrative example:

1. an *inference-based* mapping, by employing majority voting over the set of estimated labels for the windows that constitute each target recording, or,

2. a *data-driven* mapping, by utilising latent features generated by the source model on consecutive windows of a target recording to learn new models at the target-timescale directly.

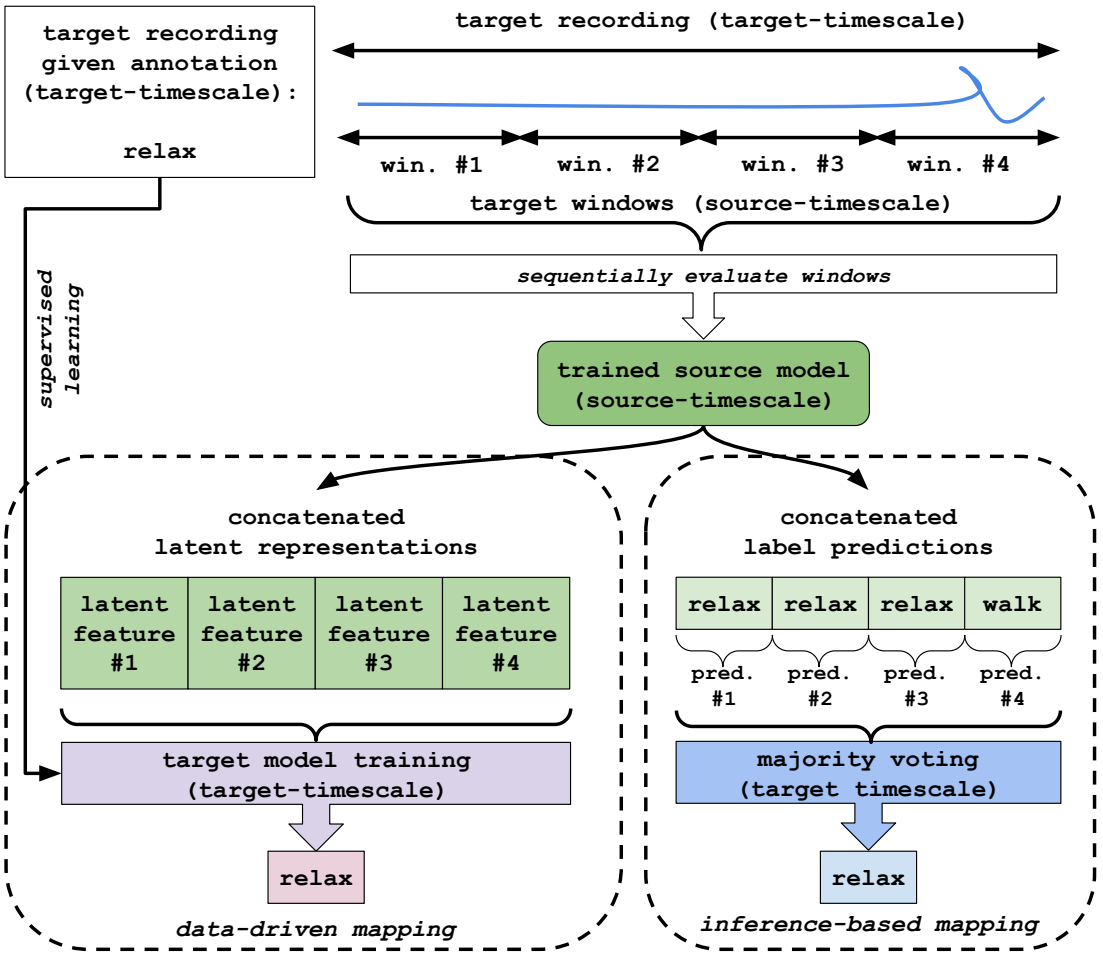


Figure 6.4: Estimation and mitigation of *timescale mismatch* - an illustrative example.

Example of *timescale mismatch*, where the frequency of annotations for the source dataset is 4× higher than in the target dataset. We can establish a *timescale mapping* utilising the sequence of latent representations (data-driven mapping) or the sequence of predicted labels (inference-based mapping) generated by a source model for consecutive windows of size source-timescale of a target recording.

Inspecting the *dataset alignment test*: if the source model learns to generalise on the target timescale and the target model learns robustly from the target

dataset, we consider the mapping effective, and source and target domains are aligned under the generated mapping. However, suppose the source model can still not learn to generalise on the target timescale despite the target model learning robustly from the target dataset. In that case, the two datasets may have a *label definition mismatch* that hinders model performance due to the differences in underlying activity definitions.

With a *hierarchy*, a *label mapping*, and a *timescale mapping* in place, we will explore in the following section how to address a *label definition mismatch* across two datasets.

6.3.3 Estimation and mitigation of *label definition mismatch*

Dataset annotation is highly dependent on how each class is defined. Different definitions of the same class label across datasets lead to *label definition mismatch*. In datasets collected in controlled environments, strict rules define each class's characteristics, which may vary among controlled datasets of the same type. For example, a sample in a HAR dataset may be considered "walking" if there is a continuous walking pattern across the recording. In contrast, samples may be annotated in another dataset as "walking" when the user performs said activity for at least 80% of a recording. In real-world data, such rules are more challenging to establish, as the participating users may have different understandings of the meaning of a label; in some cases, however, users may exhibit a shared understanding of a class even within a crowdsourced dataset, e.g., all participants may agree that they are "walking" when they perform the activity for at least 50% of the recording's length.

With *label mapping* and *timescale mapping* approaches in place to mitigate the *label content mismatch* and *timescale mismatch* across the two datasets, we introduce the notion of *purity score* to resolve such *label definition mismatches* between a source and a target dataset. We follow a similar process to mitigating the *timescale mapping*, introduced in Section 6.3.2. We create windows of size source-timescale for each recording in the target dataset. We then evaluate each of

these windows sequentially on the source model at the source-timescale, generating a sequence of predictions for each target recording. Finally, we calculate the number of predicted labels in the resulting sequence that agree with the target recording's given annotation (or its assigned higher-level annotation in the two datasets' shared semantic space after *label mapping*, if this was necessary), i.e., the number of source-timescale windows for which the predicted label is identical to their given annotation in the target dataset; this measurement is the *purity score*, p, for this target recording, $p \in \{0, N\}$, where N is the number of source-timescale-length windows that are needed to represent a target recording.

Fig. 6.5 presents an illustrative example. Having calculated the purity scores

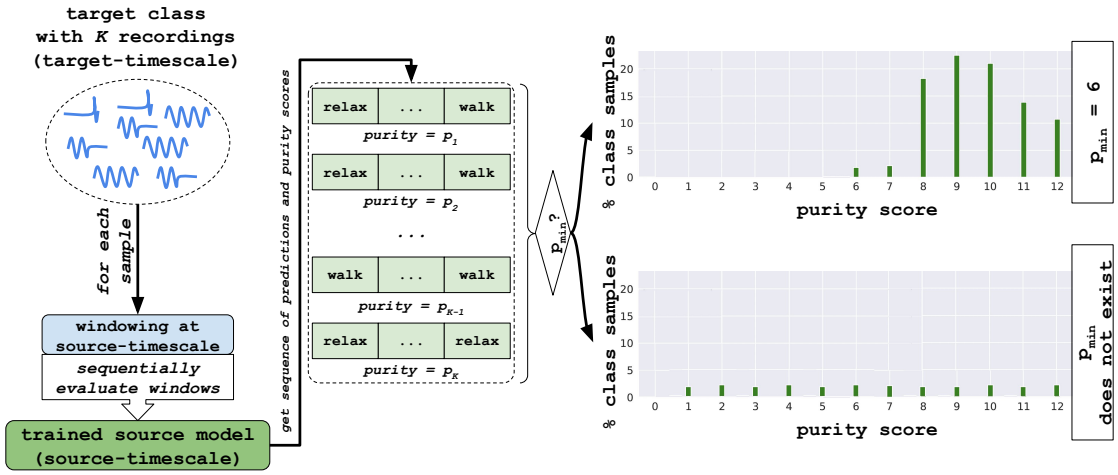


Figure 6.5: Estimation and mitigation of *definition mismatch* - an illustrative example.

Example *label definition mismatch* mitigation mechanism. After windowing target dataset recordings to windows of size source-timescale, we sequentially evaluate the windows that constitute each recording using the trained source model and obtain a sequence of estimated labels for each recording. The *purity score* is then calculated as the number of estimated labels in the sequence of source-timescale windows of a target recording that agree with the recording's given annotation at the target-timescale. By inspecting the distribution of *purity scores* for each class in the target dataset, we identify two cases: in the first case (top right bar plot), there exists a minimum *purity score* $p_{min} = 6$ which all the class samples satisfy. In the second case (bottom right bar plot), the class samples exhibit a uniform distribution of purity scores, meaning a consistent rule that governs all samples in the class does not exist.

for all samples in each target class $c \in classLabels$ at training time, we inspect their distribution: if the distribution is skewed in such away such that there exists

a minimum *purity score* $p_{min,c}$ such that, for all the target recordings within class c, it holds that $p \geq p_{min,c}$, then there exists a *cross-domain mapping* across the source and target datasets indicating the target class label requires samples to be at least $p_{min,c}$ pure with respect to the source label definition.

To determine $p_{min,c}$ for each class $c \in classLabels$, the modelwdesigner must establish sensitivity thresholds defining the proportion of an activity's occurrence within a recording that qualifies it as correctly or erroneously annotated, denoted as T_{ca} and T_{ea}, respectively.

These thresholds function as system hyperparameters and, akin to the mapping process discussed in Section 6.3.1, can be flexibly chosen based on the designer's perspective or the specific application requirements of the HAR system. Ideally, they should reflect evidence from the source domain (e.g., if walking typically occurs for at least 80% of a recording's duration in the source dataset, then $T_{ca} = 0.8$) or be guided by logical reasoning.

However, as the goal of the cross-domain mapping is to reveal consistent patterns in the target domain that may slightly differ from the source domain, rigidly transferring source domain thresholds to the target domain may obscure underlying yet consistent patterns within the latter (e.g., walking samples consistently occupying at least 70% of a recording's duration). Therefore, a more adaptable approach to defining the minimum valid $p_{min,c}$ for each class is preferable. In this work, we assume an activity recording is well-annotated (i.e., not mislabelled) if the activity occurs for at least half of its duration, leading to the selection of $T_{ca} = T_{ea} = 0.5$ and, consequently, $p_{min,c} > 0.5$ for all $c \in classLabels$.

Once these thresholds are set, $p_{min,c}$ is identified within the purity score distribution of each class c. The existence of $p_{min,c}$ requires the distribution to exhibit a bell-like shape, indicative of a normal distribution rather than a uniform onw, and to be skewed above the selected threshold T_{ca}.

The identification of $p_{min,c}$ for each class can be achieved using various techniques, including visual inspection of purity score distributions, detecting abrupt changes

in the distribution's slope (which signal the presence of the "bell"), or employing statistical measures such as skewness.

Regardless of the method used, should such $p_{min,c}$ values (determined during training) exist, they should be embedded in the decision layer of the target model to assess whether incoming samples at inference time are sufficiently pure to be assigned to a specific class. While purity scores are unavailable at inference time due to the absence of annotations, an alternative approach is to evaluate class votes across consecutive windows of incoming data forming a target recording. If the percentage of votes for a class c exceeds its purity threshold $p_{min,c}$, the target recording is assigned to that class as its nearest match. Conversely, if no class meets this criterion, the sample is deemed "ambiguous" and flagged for further inspection by the system designer.

The existence of $p_{min,c}$ reflects the classifier's sensitivity in accurately classifying data that may contain some noise while still predominantly aligning with the assigned annotation. Moreover, it facilitates the identification of subsets within class c in the target dataset, distinguishing between "trustable", "polluted", or "mislabelled" samples.

In this case, the *dataset alignment test* is primarily guided by whether an underlying rule for each class in the incoming target data can be established based on the purity score distributions, i.e., whether we can identify a $p_{min,c}$ for each class c such that $p_{min,c}$ adheres to the constraints discussed above. Incorporating $p_{min,c}$ at the decision layer of the source model as discussed above, alongside the *timescale mapping* rule can allow source-model generalisation on the target dataset at inference time, and allows for the semantic understanding of incoming data, supporting the notion of explainability.

If such a rule cannot be established, we require further insight into the dataset, focusing on inspecting the "ambiguous" and "mislabelled" samples. To achieve this, we further inspect the *label reliability mismatch* across the two datasets to identify potential variabilities within the target dataset itself.

6.3.4 Estimation and mitigation of *label reliability mismatch*

Suppose none of the above mitigation mechanisms enables the source model to generalise to the target dataset or a target model to learn on the target dataset robustly. In that case, the target dataset may not be as reliably annotated as the source dataset, leading to *label reliability mismatch*. This may result from variability over label definition across the users, i.e., no rule is consistently obeyed by all users in the dataset, or human-error misannotations may exist.

To address this challenge, we propose a purity-based *target-domain mapping* approach to dataset understanding. Using the *purity* definition introduced to resolve the *label definition mismatch*, we identify the subset of the target dataset that is consistent in its *label definition* with the source dataset. To achieve this, we systematically eliminate the samples with the lowest purity within each class, training a classifier using each refined subset. This process will be iterated, commencing from purity level 0 and advancing up to purity level N. If, during this process, we identify a p_{min} that defines an underlying rule for part of the dataset, we can iterate through the *label definition mismatch* module to refine the cross-domain mapping. With this process, we achieve data cleaning through a *conditional trust dataset curation*, which identifies the subset of the target dataset that is *explainable* and can be trusted through the scope of the source dataset.

While this method effectively distinguishes between reliable and noisy samples, whether due to "polluted" recordings, where the activity denoted in the recording's annotation is not performed consistently enough to be labelled by the classifier in the same way, or due to erroneous annotations, it also reduces the overall dataset size. This reduction may introduce challenges such as overfitting and limit opportunities for techniques such as domain adaptation or active learning.

Rather than discarding unreliable data, an alternative approach would be to leverage the extracted semantic information regarding activity patterns within polluted subsets (which is one of the key contributions of this work) to improve data quality. This could involve relabelling misclassified samples with the most

appropriate class label or employing active-class learning to identify patterns within noisy samples as potential novel activities or distinct variations of existing activities.

Having resolved all levels of mismatches, the two datasets are now alignable. This means that class predictions can be achieved at source and target timescales, and the datasets can be joined for data augmentation or continual learning.

In the following section, we will present the implementation details supporting our proposed methodology's experimental evaluation.

6.4 Implementation details

In this section, we present details regarding our experimental setup, that allow for the evaluation of our proposed approach.

6.4.1 Dataset selection

Our proposed framework aims to identify label-based and time-based mappings across two datasets of similar semantic content in order to enable their semantic alignment in a shared semantic space, where generalisation on and learning from incoming data can be effectively implemented. In this work, we utilise the **watchHAR** dataset, introduced in Chapter 3.4, as the controlled (source) dataset, a dataset of 17 activities carefully annotated at 5.12-sec windows, and the **ExtraSensory** dataset as the crowdsourced (target) dataset, a dataset collected in-the-wild and self-annotated by the data collection participants at 20-sec recordings.

As in this thesis, we focus our analysis on wrist-attached smartwatches, and to further ensure no noise arises across the two datasets from differences in sensor type or device placement, we only select the subset of each dataset that corresponds to smartwatch accelerometer data, recorded at 100 Hz at the **watchHAR** and 25 Hz at the **ExtraSensory** dataset. As such, the data in the source and target domains are recorded using the same sensor modality.

This chapter is concerned with the identification of semantic mappings across datasets that contain *semantically similar* class labels. As the selected **watchHAR** and **ExtraSensory** datasets do not strictly contain such classes, we will only use a

curated subset of 15 activity labels released with the **ExtraSensory** dataset, along with the classes defined in the **watchHAR** dataset. The 15 selected activities are summarised in Fig. 6.6, with the semantic similarities of each group of activities colour-coded and represented by a higher-level activity label.

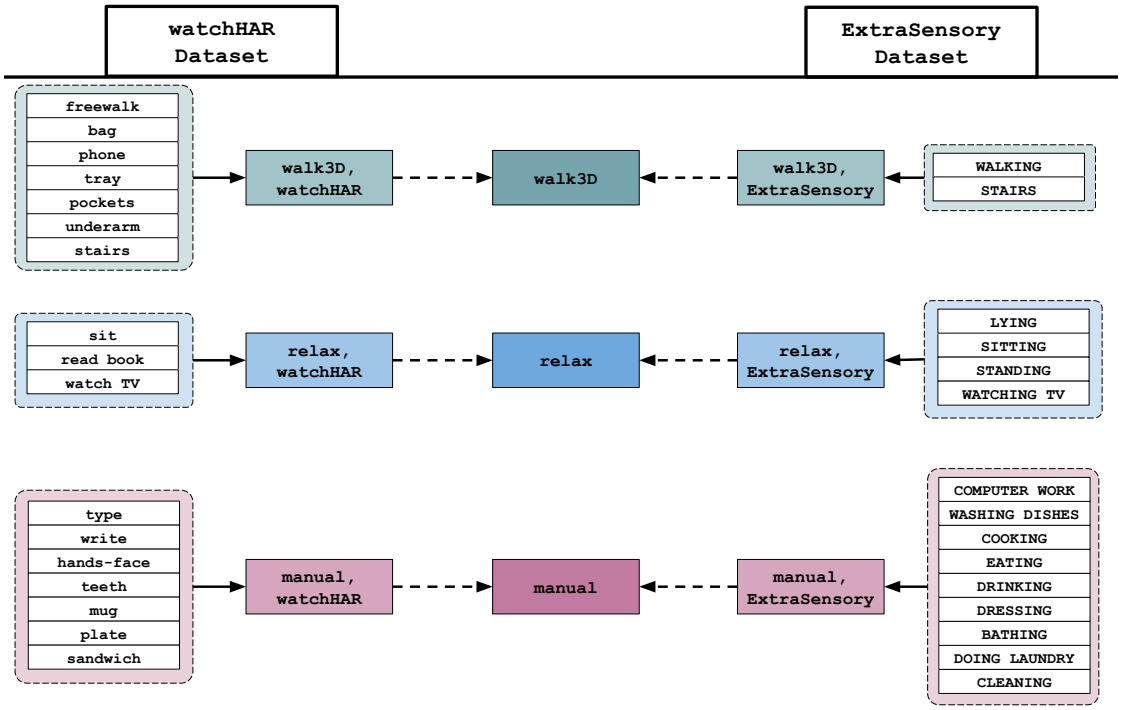


Figure 6.6: Selected subset of semantically similar activities in the **watchHAR** and **ExtraSensory** datasets, summarised in a two-level semantic inverse-hierarchy.

There are 17 activities in the source **watchHAR** dataset. We have identified 15 activities in the target **ExtraSensory** dataset that are identical or semantically similar to the activities in the source dataset. Class labels across the two datasets that exhibit semantic similarities are colour-coded with the same colour and are structured in an inverse-hierarchy under a representative class label at a higher level of semantic abstraction.

6.4.2 Feature selection

While the **watchHAR** dataset is released as raw accelerometer recordings, the **ExtraSensory** dataset is released both in the form of raw recordings and in a featurised form with a total of 255 handcrafted features, 46 of which have been extracted from the smartwatch accelerometer sensor. To our knowledge, existing literature has only used the featurised **ExtraSensory** dataset; as such, we consider

it necessary to use the featurised version of the data to establish baseline model performances for the **ExtraSensory** dataset. However, we also investigate model performance on the raw data recordings, in line with the **watchHAR** dataset. For the remainder of this work, we will denote the two versions of the **ExtraSensory** dataset as **rawWatch-ExtraSensory** and **featWatch-ExtraSensory** for the raw and featurised form of the smartwatch accelerometer data, respectively.

6.4.3 Data preparation

The **watchHAR** dataset and the **rawWatch-ExtraSensory** dataset are preprocessed in a similar way. The data are linearly interpolated to account for missing timestamps. The data in **rawWatch-ExtraSensory** are then upsampled from 25 Hz to 100 Hz to match the sampling frequency of the **watchHAR** data. Then, a moving median filter is utilised for signal denoising in both datasets, followed by a high-pass Butterworth filter with a cut-off frequency of 0.3 Hz to remove the gravity component. We then filter the data with a low-pass Butterworth filter with a cut-off frequency of 20 Hz, the band of frequencies of human activities, to remove any high-frequency noise. The data are then bounded, normalised to zero mean and unit standard deviation, and windowed to 5.12-sec windows with 75% overlap. For the **rawWatch-ExtraSensory** dataset, the original 20-sec length recordings correspond to twelve consecutive 5.12-sec windows due to the 75% overlap.

The data in **featWatch-ExtraSensory** are normalised by subtracting the mean and dividing by the standard deviation for each feature.

To ensure meaningful comparison of the two versions of the **ExtraSensory** dataset, we need to ensure that the two datasets correspond to raw and featurised forms of the same samples: we discard any samples that have missing (NaN) features, as well as samples that have invalid raw datafiles, arising from missing data, inconsistent timestamps, or undocumented recording duration. At the end of this process, **rawWatch-ExtraSensory** and **featWatch-ExtraSensory** will contain precisely the same samples in raw and featurised form, respectively.

6.4.4 Model selection

In this work, we utilise a combination of deep and shallow classification models.

6.4.4.1 Source-model learning and generalisation on target data

To train a source-model on the **watchHAR** dataset, we employ the wavelet transform to transform the raw **watchHAR** data to a representative 2D scalogram, similar to our approach in Chapter 5.7. Having previously used a CNN-LSTM architecture, we will now use the ResNet9 model [397], a well-known architecture which can help to avoid issues such as overfitting due to its relatively small size. We denote the ResNet9 model trained with wavelet 2D scalograms as inputs as wavResNet9. Having preprocessed the watchHAR data as described in Section 6.4.3, we compute the DWT over the 5.12-sec length windows across seven scales, resulting in features of shape $3 \times 7 \times 512$, i.e., a 7×512 scalogram for each of the three accelerometer channels, as 512 wavelet coefficients are calculated for each of the 7 scales utilised to calculate the DWT; for DWT, each scale corresponds to a frequency band within the signal. Then, as the typical ResNet9 model expects inputs of shape 256×256, we downsample the resulting scalograms from 512 to 256 across the time dimension and upsample from 7 to 256 across the scale dimension. Downsampling in the coefficient dimension is performed by selecting every second sample of the original 512-dimensional features, while the nearest-neighbour mechanism for missing is chosen for upsampling in the scale dimension. Finally, we normalise the training data to zero mean and unit standard deviation.

This results to 57353 recordings for the ExtraSensory dataset, each yielding 12 windows of 5.12-sec duration. With 1 user left out as the testing set (with 11462 recordings), the recordings of the remaining users are split among the training test and validation set. 80% of the recordings of each user is selected for training, with the remaining 20% used for validation. This results in 40142 recordings (481704 5.12-sec windows) in the ExtraSensory training test, 5749 recordings (68988 5.12-sec windows) in the ExtraSensory validation test and 11462 recordings (137544 5.12-sec windows) in the validation set.

To evaluate the **ExtraSensory** data on the trained source model and inspect its generalisation capabilities to the new target domain, we also preprocess the **rawWatch-ExtraSensory** subset in the same way, utilising the wavelet transform.

6.4.4.2 Target-model learning

Inspecting the generalisation capabilities of a source model on the target dataset is only one component indicating dataset alignment; the second one is the learning capacity and robustness of models trained on the target dataset itself.

For this task, we follow the recommended classification approach by the curators of the dataset [393]; each label is modelled individually using binary logistic regression (BLR). This is a valuable baseline, as it demonstrates the separability of each activity from all others. Then, we generalise to the direct multiclass classification approach using a multiclass logistic regressor (with the one-vs-rest criterion). We use several other shallow classifiers, namely, decision trees (DT), random forests (RF), support vector machines with both linear (SVM_{lin}) and radial basis function (SVM_{rbf}) kernels, gradient boosting classifier (GBC) and Gaussian Naive Bayes (GNB), as recommended in [40].

With the exception of the binary logistic regressor, which is only used to train a baseline model on the **featWatch-ExtraSensory** dataset, all other classifiers are used to train target models by utilising latent encodings of the **rawWatch-ExtraSensory** dataset, generated by the above-defined source model. We examine the following latent features:

- *Latent*: the latent representations of the **wavResnet9** model, that are the input vectors to its final fully-connected layer, concatenated sequentially for the twelve source-timescale sized windows that constitute a target recording.

- *Logits*: the output vectors of the final fully-connected layer of the **wavResnet9** model, concatenated sequentially for the twelve source-timescale sized windows that constitute a target recording.

- *Probs*: the softmax probabilities of *Logits*, concatenated sequentially for the twelve source-timescale sized windows that constitute a target recording.
- *labelsSeq*: the generated sequence of estimated labels for the twelve source-timescale-sized windows that constitute a target recording.
- *labelsFreq*: the frequency of each class in labelsSeq.

Having decided on the source and target datasets to inspect and all the basic data preprocessing and model definitions in place, we are now in a position to evaluate our proposed approach, **SeMEDA**.

6.5 Evaluation

In this section, we will discuss the experimental details of our proposed approach. We will evaluate each component of our framework on a source-target dataset pair, the controlled **watchHAR** dataset and the crowdsourced **ExtraSensory** dataset, respectively, preprocessed as indicated in Section 6.4.4.2.

Let us inspect the two datasets. As demonstrated in Fig. 6.6, the class labels are not identical across the datasets; hence there exists *label content mismatch*. Furthermore, the **watchHAR** dataset is annotated over 5.12-sec windows, while the **ExtraSensory** dataset is annotated at 20-sec windows, leading to *timescale mismatch*. As the datasets were collected independently, the underlying rules that govern each activity class are likely different across the datasets, thus causing *label definition mismatch*. Finally, as the **ExtraSensory** dataset was collected in real-world conditions from various independent users, the task perception across the participants likely varies for the same activity label, as well as that there is an increased possibility for annotation errors, highlighting potential *label reliability mismatch*.

Based on the above, we expect it will be necessary to estimate and mitigate all four levels of semantic mismatch between the **watchHAR** and **ExtraSensory** data

to achieve dataset alignment. In the remainder of this section, we will sequentially evaluate the identified mismatches.

We perform one-subject-out evaluation, utilising one left-out subject as the test set. From the remaining users, 80% of each user's data are utilised for training, and the remaining 20% for validation. The watchHAR dataset contains 21 subjects, and the ExtraSensory dataset 60 subjects. We only utilise the accelerometer data, as no gyroscope data are available in the ExtraSensory dataset. All data are preprocessed as discussed in Section 6.4.3.

We compare our proposed approach with the DeepCoral [287] method for feature alignment between source and target domains, in terms of handling *label content mismatch*. We also inspect our proposed approach's performance by utilising sample-relabelling, as this is indicated by the calculated purity scores, to train models directly in the target domain.

6.5.1 Estimation and mitigation of *label content mismatch*

In this section, we will evaluate the *hierarchy generation and label mapping* approach we introduced in Section 6.3.1 to resolve the *label content mismatch*.

6.5.1.1 Hierarchy generation and label mapping

Let us revisit Fig. 6.6, which summarised the class labels of the source and target datasets. We identify three abstracted class labels that can encompass all the semantic similarities among the classes in the two datasets: `walk3D`, incorporating walk events within or across floors (normal walking and going up/down stairs, respectively); `relax`, for all static, low-energy activities; and `manual`, for all activities that require active usage of the user's hands, without walking events. As such, we must establish models to classify samples into these three categories.

6.5.1.2 Dataset alignment test #1

To examine whether the *generated hierarchy* and *label mapping* suffice for dataset alignment, we perform the following alignment test: (1) can a source model trained on the **watchHAR** dataset, mapped in the shared high-level semantic space, generalise

on source-timescale-sized windows of incoming samples from the **ExtraSensory** dataset, and (2) is it possible for a new target model to be trained robustly on the **ExtraSensory** dataset? To perform this test, we must establish a robust source domain baseline.

Baseline: learning a source-model from the watchHAR dataset As discussed in Section 6.4.4, we employ a wavResNet9 classifier with three output classes to train the source model from the raw **watchHAR** data over source-timescale windows of 5.12-sec. We denote the learned source model as C_w. The results are summarised in Table 6.1. The model is robust when evaluated on the **watchHAR** test set, yielding a balanced accuracy of over 90% for all classes.

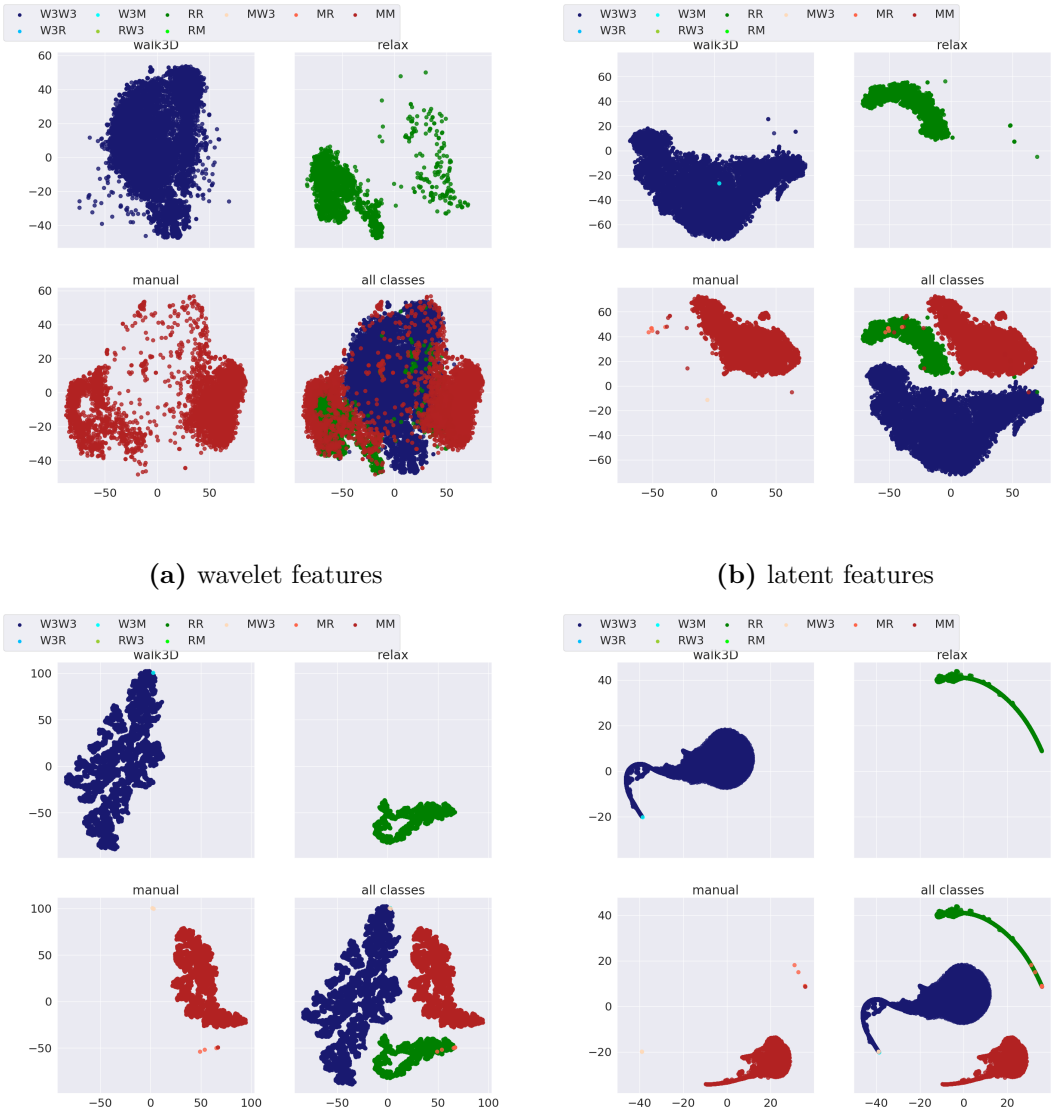
Table 6.1: Performance evaluation of the source model C_w on the **watchHAR** test set.

The source model C_w, trained on the **watchHAR** dataset, is robust when evaluated on a withheld test set, yielding a robust baseline.

label	C_w	
	per class balanced accuracy	**balanced accuracy**
`walk3D`	99.26%	
`relax`	90.12%	94.09%
`manual`	95.01%	

We also examine the separability capabilities of the trained model at four representations for the data: at the input level (wavelet features), at a latent representation (intermediate representation before the final fully-connected layer), at the logits level (output of fully-connected layer) and the output level (estimated softmax probabilities). Fig. 6.7 demonstrates how the model improves class separability from the input to the output level for all classes in the shared semantic space.

With this baseline and observations in place, let us inspect the two components of the *dataset alignment test*: *source-model generalisation* and *target-model learning*.



(a) wavelet features

(b) latent features

(c) logits features

(d) softmax probabilities features

Figure 6.7: TSNE visualisation for training samples in **watchHAR**.

TSNE visualisation for the **watchHAR** dataset, using as features (a) the input-level wavelet features, (b) a latent representation from the wavResNet9 model, ahead of its final fully-connected layer, (c) the model's logits, i.e., the output of its final fully-connected layer and (d) the output softmax probabilities. We observe that classes form dense clusters that become notably more separable as we utilise deeper feature representations. Notation: XY := (true X, estimated Y), where X,Y in {W3 - walk3D; R - relax; M - manual}.

Generalisation test: source-model generalisation on the ExtraSensory dataset at source-timescale Having trained a robust source model C_w, we inspect its generalisation capabilities on incoming real-world data, simulated by the **rawWatch-ExtraSensory** dataset, preprocessed as described in Section 6.4.3. Windowed at 5.12-sec windows with 75% overlap, each 20-sec recording in the dataset is evaluated in the form of twelve 5.12-sec independent window segments. To measure the model's generalisation performance, we assume that the given annotations in the target dataset are uniformly applied over the recording length. Hence, all 5.12-sec windows also have the same annotations as the original 20-sec recording. The results are summarised in Table 6.2 under the notation $C_{w,evalWin}$. Evidently, the source model, effectively trained on **watchHAR** struggles to identify

Table 6.2: Performance evaluation of $C_{w,evalWin}$ model on the **rawWatch-ExtraSensory** test set.

The source model C_w, trained on the **watchHAR** dataset, is unable to generalise to the **ExtraSensory** dataset at source-timescale ($C_{w,evalWin}$ mode) following a *hierarchical label mapping* to resolve *label content mismatch*. This is particularly prevalent for the `walk3D` and `manual` classes; hence, we expect more semantic mismatches among those classes. Instead, the `relax` class is robust in its predictions.

label	$C_{w,evalWin}$	
	per class balanced accuracy	**balanced accuracy**
`walk3D`	34.93%	
`relax`	82.57%	43.57%
`manual`	15.52%	

the **rawWatch-ExtraSensory** samples with respect to their intuitively mapped annotations. In particular, the more "active" classes `walk3D` and `manual` are prone to misclassification, while on the contrary, the `relax` class is robust in its predictive accuracy.

Let us visually examine the predictions of the C_w model at source-timescale. Fig. 6.8 revisits the TSNE visualisations for the **watchHAR** dataset at the softmax probabilities representation level, overlaying the **rawWatch-ExtraSensory** samples on its TSNE embedding in 2D space. As can be seen in the visualisations,

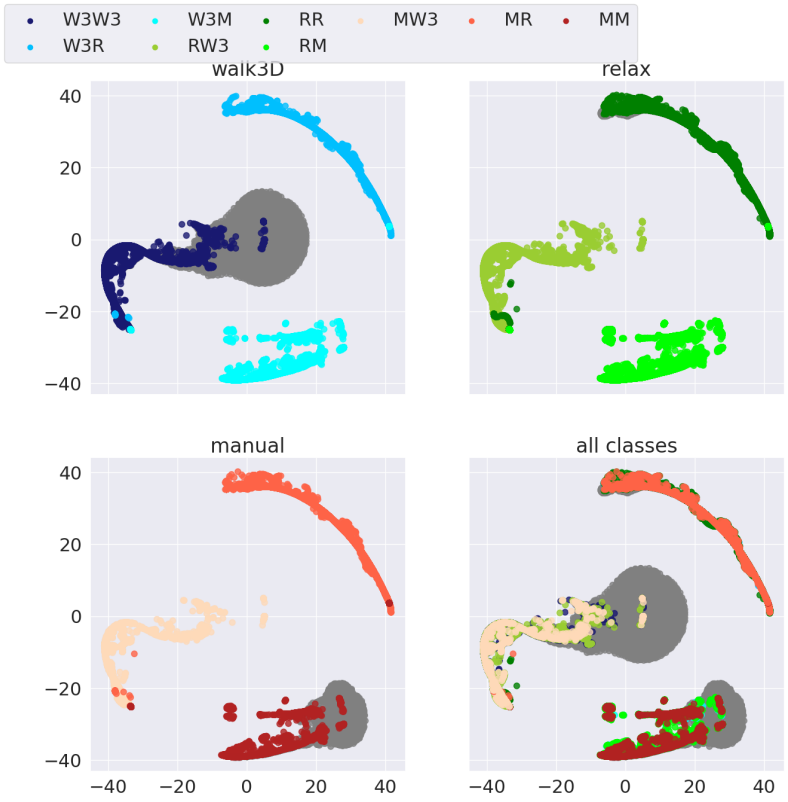


Figure 6.8: TSNE visualisation for training samples in **rawWatch-ExtraSensory** at softmax probabilities representation.

TSNE visulisation for the **rawWatch-Extrasensory** dataset at using as features the source model's output softmax probabilities, overlayed over the TSNE cluster visualisations of the **watchHAR** dataset in Fig. 6.7d. Each 20-sec recording in the dataset is evaluated in 12 5.12-sec window segments (i.e., each dot on the TSNE visualisation represents a window of a **rawWatch-Extrasensory** sample). We observe that many samples seem misclassified; however, they are always placed on the embedding of their estimated cluster in the TSNE visualisation. Notation: XY := (true X, estimated Y), where X, Y in {W3 - walk3D; R - relax; M - manual}.

the 2D representation of the 5.12-sec windows is well-aligned with the TSNE cluster corresponding to their estimated labels without disrupting the decision margins, which signifies high similarity of the evaluated windows with the samples of their estimated class in the **watchHAR** dataset. This suggests that the incoming activities may be composite, contrary to their assigned label, hence leading to *label definition mismatch*. We will investigate this in Section 6.5.3.

To determine whether the poor performance of the source model in the target domain is solely attributable to user and device heterogeneity, we compare our proposed approach with Deep Coral, an unsupervised domain adaptation method. For a direct comparison, we train and evaluate the source model both with and without Coral loss, ensuring that all other parameters remain unchanged.

To achieve robust training in both conditions, we adjusted certain parameter values from those used in the results reported earlier and later in this section (such as learning rate and number of epochs for training). As a result, the findings presented in Table 6.3 are directly comparable only with one another and should not be contrasted with the results elsewhere in this section.

Although Deep Coral operates in an unsupervised manner, aligning source and target feature spaces without leveraging target labels, it proves insufficient for effectively adapting models to the target domain.

Our SeMEDA analysis reveals that the dataset suffers from poor annotation, including both ambiguous and mislabelled samples. Consequently, even if feature spaces are aligned, misclassification persists when evaluating against target labels.

The limited generalisability of Deep Coral underscores the necessity of not only transferring knowledge across domains but also identifying which data subsets are truly alignable.

For the remainder of this section, we proceed our evaluation without Deep Coral, as it does not assist to handle the issues at hand.

Table 6.3: Performance evaluation of C_w (on a watchHAR withheld test set) and $C_{w,evalWin}$ model on the **rawWatch-ExtraSensory** test set with and without Coral loss for domain adaptation.

The source model C_w, while generalising well on the **watchHAR** dataset, is unable to generalise to the **ExtraSensory** dataset at source-timescale ($C_{w,evalWin}$ mode) even when applying Deep CORAL for domain adaptation. This is particularly prevalent for the `walk3D` and `manual` classes; hence, we expect more semantic mismatches among those classes. Instead, the `relax` class is robust in its predictions.

label	C_w			
	per class balanced accuracy		**balanced accuracy**	
	w/o coral	w. coral	w/o coral	w. coral
`walk3D`	93.30%	95.71%		
`relax`	86.96%	81.43%	89.03%	90.66%
`manual`	91.08%	91.10%		
label	$C_{w,evalWin}$			
	per class balanced accuracy		**balanced accuracy**	
	w/o coral	w. coral	w/o coral	w. coral
`walk3D`	30.29%	34.67%		
`relax`	92.03%	92.13%	41.29%	39.69%
`manual`	9.05%	9.10%		

Learning test: target-model learning on the ExtraSensory dataset at target-timescale After inspecting the generalisation capabilities of the source model to the target dataset, here we will investigate the capacity and robustness of models trained on the **ExtraSensory** dataset. As the primary release of the **ExtraSensory** dataset is in its featurised form, we will begin by comparing several baselines on the **featWatch-ExtraSensory** dataset. We investigate the performance of several classifiers recommended in literature [40, 393]. We denote these as $C_{e,m}$ where $m \in \{BLR, MLR, DT, RF, SVM_{lin}, SVM_{rbf}, GBC, GNB\}$ denoting the binary linear regressor, multiclass linear regression, Decision Tree, random forest, SVM (linear kernel), SVM (radial basis function kernel), gradient boosting and gaussian naive bayes models, respectively. The results are summarised in Table 6.4. A few interesting observations are in order. First, as in the generalisation test conducted above, the more active classes `walk3D` and `manual`

Table 6.4: Performance evaluation of $C_{e,m}$ models

The target models $C_{e,m}$, $m \in \{BLR, MLR, DT, RF, SVM_{lin}, SVM_{rbf}, GBC, GNB\}$ denoting the binary linear regressor, multiclass linear regression, Decision Tree, random forest, SVM (linear kernel), SVM (radial basis function kernel), gradient boosting and gaussian naive bayes models, respectively, are not capable to train robustly on the **featWatch-ExtraSensory** dataset.

model	**label**			
	per class balanced accuracy			**balanced accuracy**
	`walk3D`	`relax`	`manual`	
C_{e,BLR^S}	79.07%	70.28%	64.11%	-
$C_{e,MLR}$	50.21%	81.01%	25.42%	55.99%
$C_{e,DT}$	42.29%	71.38%	34.93%	55.12%
$C_{e,RF}$	47.53%	70.71%	33.20%	54.32%
$C_{e,SVM_{lin}}$	50.70%	78.61%	28.75%	55.03%
$C_{e,SVM_{rbf}}$	46.07%	77.38%	28.72%	55.03%
$C_{e,GBC}$	47.83%	75.26%	32.65%	57.04%
$C_{e,GNB}$	46.86%	84.73%	17.02%	52.36%

are more prone to reduced performance compared to the `relax` class. Furthermore, performance is significantly affected for these classes when moving from binary to multiclass models. As such, the selected models cannot draw an accurate decision boundary across the classes for the multiclass case.

To investigate this further, we inspect the features released in **featWatch-ExtraSensory**. Fig. 6.9 illustrates the 2D TSNE visualisation of the training samples from **featWatch-ExtraSensory**. The samples from the different classes are not separable in the released smartwatch feature space, impeding the learning process of the various models. As a result, it is essential to investigate different feature representations for the **ExtraSensory** dataset, generating new features from the **rawWatch-ExtraSensory** dataset directly.

In summary, the **featWatch-ExtraSensory** dataset is challenging to establish robust classification models. We have demonstrated this holds partly due to the separability of the released features; in the following section, we will attempt to train the target models from **rawWatch-ExtraSensory** dataset. As the source model C_w is already capable of identifying patterns from raw data, we will attempt

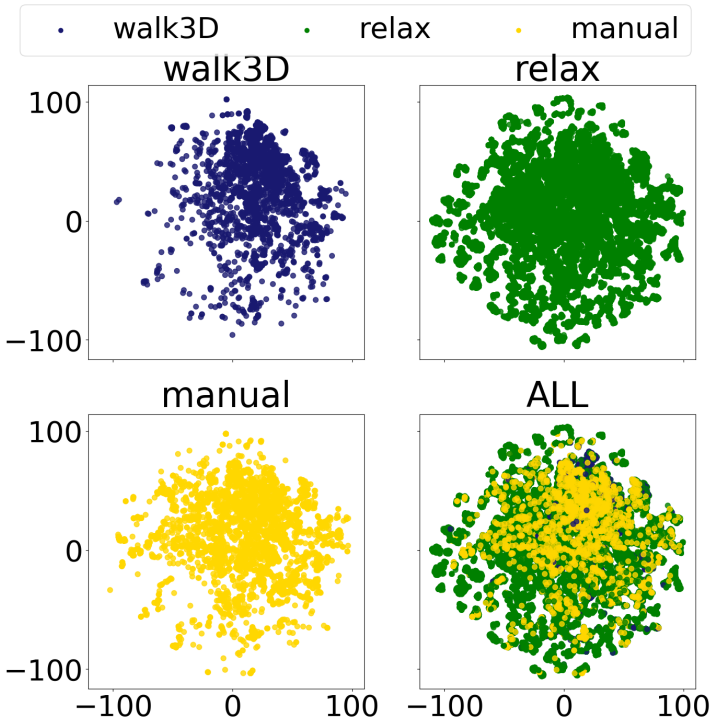


Figure 6.9: TSNE visualisation for training samples in **featWatch-ExtraSensory**.

TSNE visulisation for the **featWatch-ExtraSensory** dataset. We observe that classes are not separable in this representation.

to investigate the **ExtraSensory** data through the lens of the source model that was trained on **watchHAR**, which was very carefully and accurately labelled under controlled lab conditions. To achieve this, we will attempt to *transfer* the knowledge embedded in C_w to the target models. However, as C_w makes predictions at source-timescale, a mitigation mechanism for *timescale-mismatch* is required. We will investigate this in the following section.

Overall, we observe that, as expected, mitigating the *label definition mismatch* by projecting the classes from both datasets in a shared semantic space is not sufficient for the alignment of the **watchHAR** and **ExtraSensory** datasets. In the following section, we will investigate how addressing the *timescale mismatch* can assist towards accomplishing the dataset alignment goal.

6.5.2 Estimation and mitigation of *timescale mismatch*

In the previous section, we established a *label mapping* from the **watchHAR** and **ExtraSensory** dataset to a shared, abstracted semantic space, aiming to overcome their *label content mismatch*. In this space, it is possible for a source model trained on **watchHAR** to be used to provide activity labels over windows of incoming **ExtraSensory** data; recall that the **watchHAR** dataset is annotated over 5.12-sec windows (source-timescale), while the **ExtraSensory** labels are assigned over 20-sec recordings (target-timescale). In this section, we will investigate mechanisms to mitigate the *timescale mismatch* across the two datasets and allow for a unique class prediction for the entire length of the recordings in the target dataset.

6.5.2.1 Timescale mapping

As the source model has been trained on the **watchHAR** dataset over 5.12-sec windows, it cannot provide activity estimates for incoming samples from the **ExtraSensory** dataset at a 20-sec window directly. At the same time, we demonstrated in Section 6.5.1 that the featurised **featWatch-ExtraSensory** dataset could not distinguish classes effectively to robustly train models at target-timescale directly.

To enable our proposed *timescale-mapping* approaches, we enable a memory buffer of twelve capable of withholding latent or output representations for the twelve consecutive 5.12-sec windows constituting a 20-sec recording of the target dataset. An *inference-based mapping* then utilises a *majority voting* mechanism over the sequence of predicted labels in the memory buffer. Alternatively, a *data-driven mapping* utilises the sequence of latent feature representations in the memory buffer to learn new target models directly. Both approaches borrow from transfer learning techniques to utilise the class separability knowledge embedded in the source model to enable predictions in the target-timescale.

6.5.2.2 Dataset alignment test #2

To examine whether the proposed *timescale mapping* suffices for dataset alignment, we perform the following alignment test: (1) can a source model trained on the **watchHAR** dataset, mapped in the shared high-level semantic space, be modified to generalise on incoming samples from the **ExtraSensory** dataset at their original target-timescale, and (2) is it possible for a new target model to be trained robustly on the **ExtraSensory** dataset? This test's generalisation and learning components correspond to inspecting the model performances of the *inference-based* and *data-driven* proposed approaches, respectively.

Generalisation test: source-model generalisation on the ExtraSensory dataset at source-timescale with *inference-based mapping* With a memory buffer of twelve enabled over the trained source model, we obtain a sequence of twelve estimated labels by sequentially evaluating the twelve windows that constitute a target recording on the source model. We define the most frequently appearing label in the sequence (*majority voting*) as the estimated label for the target recording at the target-timescale. The results are summarised in Table 6.5, denoted $C_{w,evalRec}$. Examining these results alongside Table 6.2, we observe that

Table 6.5: Performance evaluation of $C_{w,evalRec}$ model on the **rawWatch-ExtraSensory** test set.

The source model C_w, trained on the **watchHAR** dataset, is unable to generalise to the **ExtraSensory** dataset at target-timescale ($C_{w,evalRec}$ mode) following as *inference-based timescale mapping* to resolve *timescale mismatch*. This is particularly prevalent for the `walk3D` and `manual` classes; hence, we expect more semantic mismatches among those classes. Instead, the `relax` class is robust in its predictions.

label	$C_{w,evalRec}$	
	per class balanced accuracy	**balanced accuracy**
`walk3D`	47.53%	
`relax`	91.10%	45.32%
`manual`	11.80%	

the overall model performance is improved as we move from evaluating at source-timescale to evaluating at target-timescale with *majority voting*. However, the source model C_w still struggles to identify the **rawWatch-ExtraSensory** samples at target-timescale, hinting there exist further mismatches in the datasets that need to be resolved to achieve dataset alignment.

The above inference-based mapping attempts to resolve the *timescale mismatch* by postprocessing the source models' outputs, thus not requiring further training. In the data-driven mapping, we will attempt to learn models that predict directly at the target-timescale by utilising various latent features generated by the source models.

Learning test: target-model learning on the ExtraSensory dataset at target-timescale with *data-driven mapping* In the data-driven approach to resolving the *timescale mismatch*, we attempt to utilise latent representations of the target dataset generated by the source model to encode the target samples at the target-timescale directly. For each of the generated feature representations, we revisit the classifiers we examined in Table 6.4 for each generated feature representation. We denote the trained models $C_{e,mf}$ where $m \in \{MLR, DT, RF, SVM_{lin}, SVM_{rbf}, GBC, GNB\}$ denoting the multiclass linear regression, decision tree, Random Forest, SVM (linear kernel), SVM (radial basis function kernel), gradient boosting and gaussian naive bayes models, respectively, and $f \in \{Latent,\ Logits,\ Probs,\ labelsSeq,\ labelsFreq\}$, denoting the latent representations we defined in Section 6.4.4.2. The results are summarised in Fig. 6.10.

Similar to inference-based mapping, the data-driven approach establishes a way to make predictions at the target-timescale. While there is a marginal improvement in the predictive accuracy of the learned models, depending on the choice of model and feature representation, performance is not improved significantly, hinting further factors are affecting the semantic alignment of the two datasets. In the next section, we will attempt to identify and resolve the *label definition mismatch* across the two datasets.

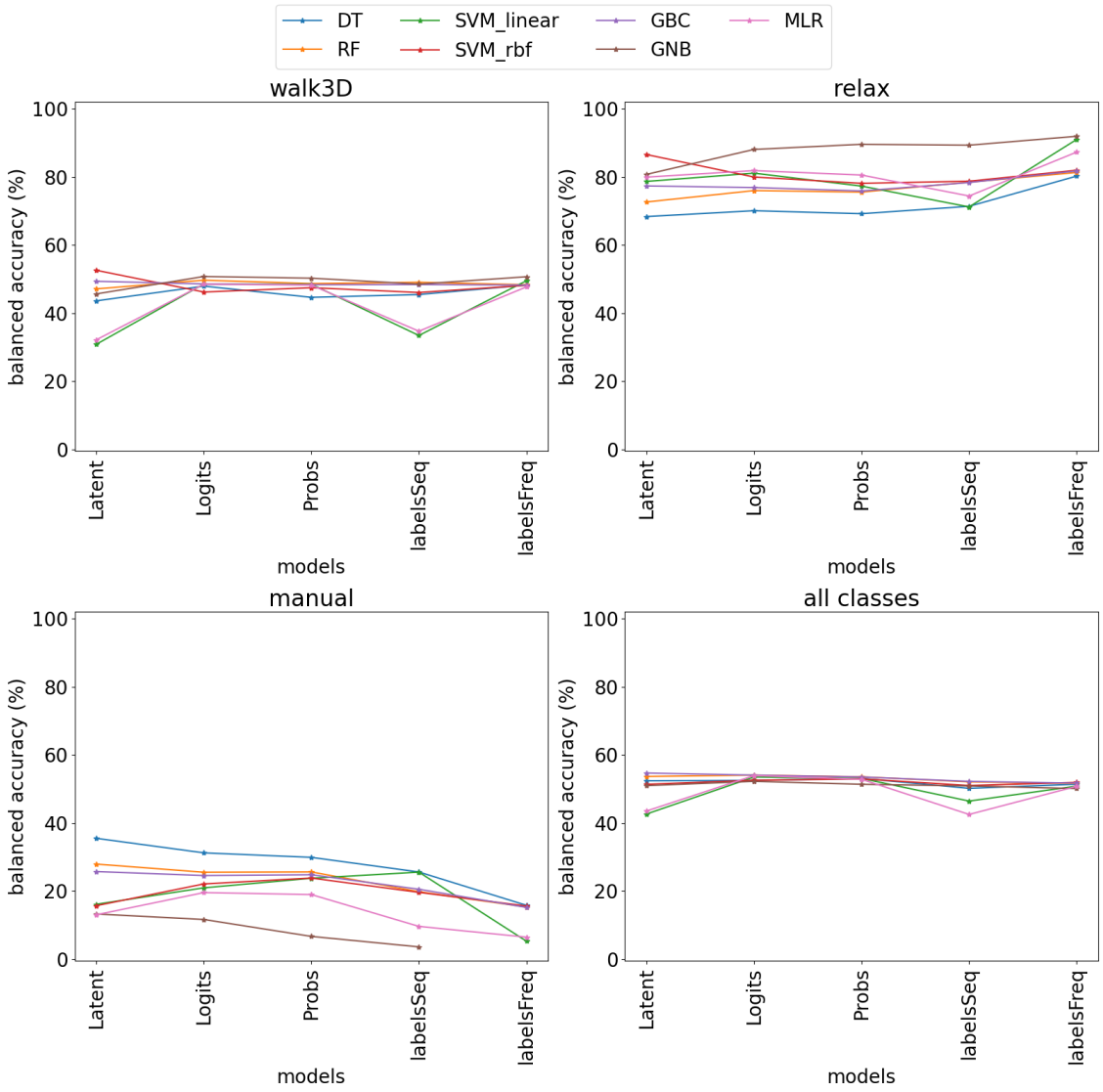


Figure 6.10: Performance analysis on the effect of *data-driven timescale mappings* at learning from various latent representations of the **rawWatch-ExtraSensory** dataset.

While there is a marginal improvement in the predictive accuracy of the learned models compared to the *inference-based timescale mapping*, summarised in Table 6.5, the target models $C_{e,mf}$, $m \in \{MLR, DT, RF, SVM_{lin}, SVM_{rbf}, GBC, GNB\}$ denoting the multiclass linear regression, decision tree, Random Forest, SVM (linear kernel), SVM (radial basis function kernel), gradient boosting and gaussian naive bayes models, respectively, and $f \in \{Latent,\ Logits,\ Probs,\ labelsSeq,\ labelsFreq\}$, denoting the latent representations, still struggle to learn robustly on the **ExtraSensory** following a *data-driven timescale mapping* to resolve *timescale mismatch*. This is particularly prevalent for the `walk3D` and `manual` classes; hence, we expect more semantic mismatches among those classes. Instead, the `relax` class is robust in its predictions.

6.5.3 Estimation and mitigation of *label definition mismatch*

In the previous sections, we demonstrated that resolving the *label content* and *timescale mismatches* is not sufficient to align the **watchHAR** and **ExtraSensory** datasets semantically. This section will examine the *label definition mismatch* across the two datasets.

6.5.3.1 Cross-domain mapping

To address this challenge, we define the *purity score* p, $p \in \{0, \ldots, 12\}$ between a target sample's given annotation at target-timescale and the sequence of estimated labels at source-timescale over its corresponding twelve 5.12-sec windows; the purity score is defined as the number of 5.12-sec windows in the recordings for which the inferred label at source-timescale agrees with their given annotation at target-timescale. Then, we inspect the distributions of samples over the purity scores for each possible outcome of the classifier, i.e., the set of true VS estimated label pairs. This can shed light on the composition of each class and how this affects the classification performance.

6.5.3.2 Dataset alignment test # 3

In this case, the *dataset alignment test* is guided by the existence of a consistent underlying rule governing each class, revealed by the inspection of the purity distribution: if there exists a purity score p_{min} such that for all samples within a class it holds that $p \geq p_{min}$, then we can consider the datasets alignable. Fig. 6.11 summarises the analysis for the samples of the **ExtraSensory** across the three classes of the shared semantic space.

A few intriguing observations merit discussion; while the `relax` class has high purity scores, only a small percentage of the `walk3D` and `manual` classes score similarly. For example, let us examine the purity scores for the `walk3D` class. While only marginally more than 10% of the samples in the `walk3D` class samples are estimated as entirely pure ($purity = 12$), more than 35% of the samples

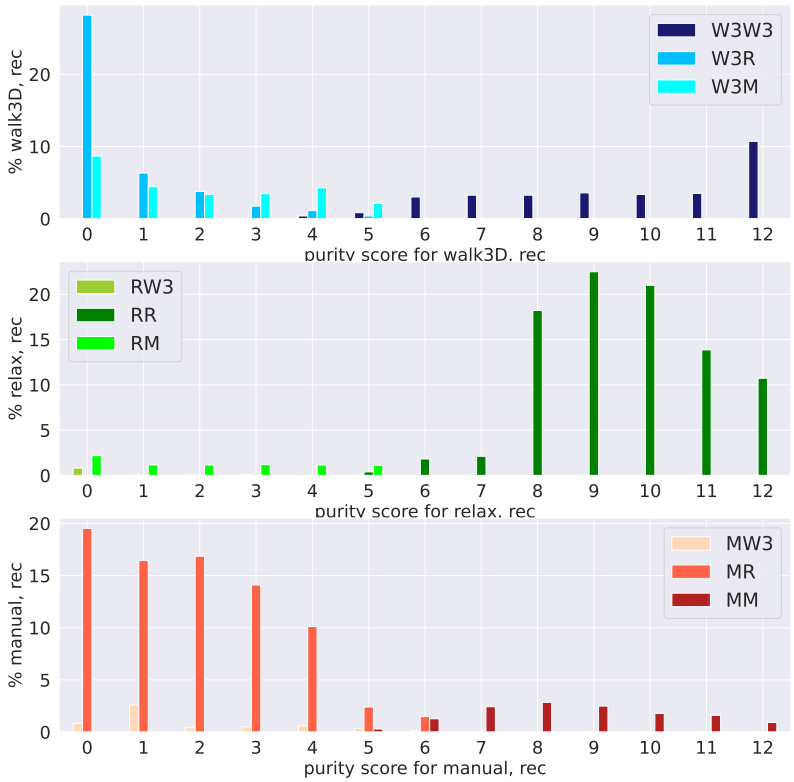


Figure 6.11: Distribution of predicted labels at real-scale based on *purity score* for each class in the shared semantic space.

Barplot visualisation depicting the model's inferences over the purity score for each class in the shared semantic space. Purity $\in [0, 12]$ is the number of windows in an extrasensory recording for which the inferred label is the same as its given annotation. Notation: XY := (true X, estimated Y), where X, Y in {W3 - walk3D; R - relax; M - manual}.

annotated as `walk3D` at target-timescale are estimated as `relax` or `manual`, with zero similarity to the `walk3D` class ($purity = 0$). The `manual` class also has a similar behaviour, with only approximately 1% of the class scoring $purity = 12$, and 20% of the class scoring $purity = 0$.

Finally, let us visually inspect a few randomly selected samples for each true-VS-estimated pair of labels. We utilise the inference-based majority voting mapping to obtain predictions at the target-timescale. As an illustrative example, we will examine the true-VS-estimated pairs of labels at GL_2. Fig. 6.12 visualises five random samples for each true-VS-estimated pair of labels.

As is demonstrated in the raw signal visualisations, the samples that are "misclassified" by our classifier (with majority voting) are largely similar to the

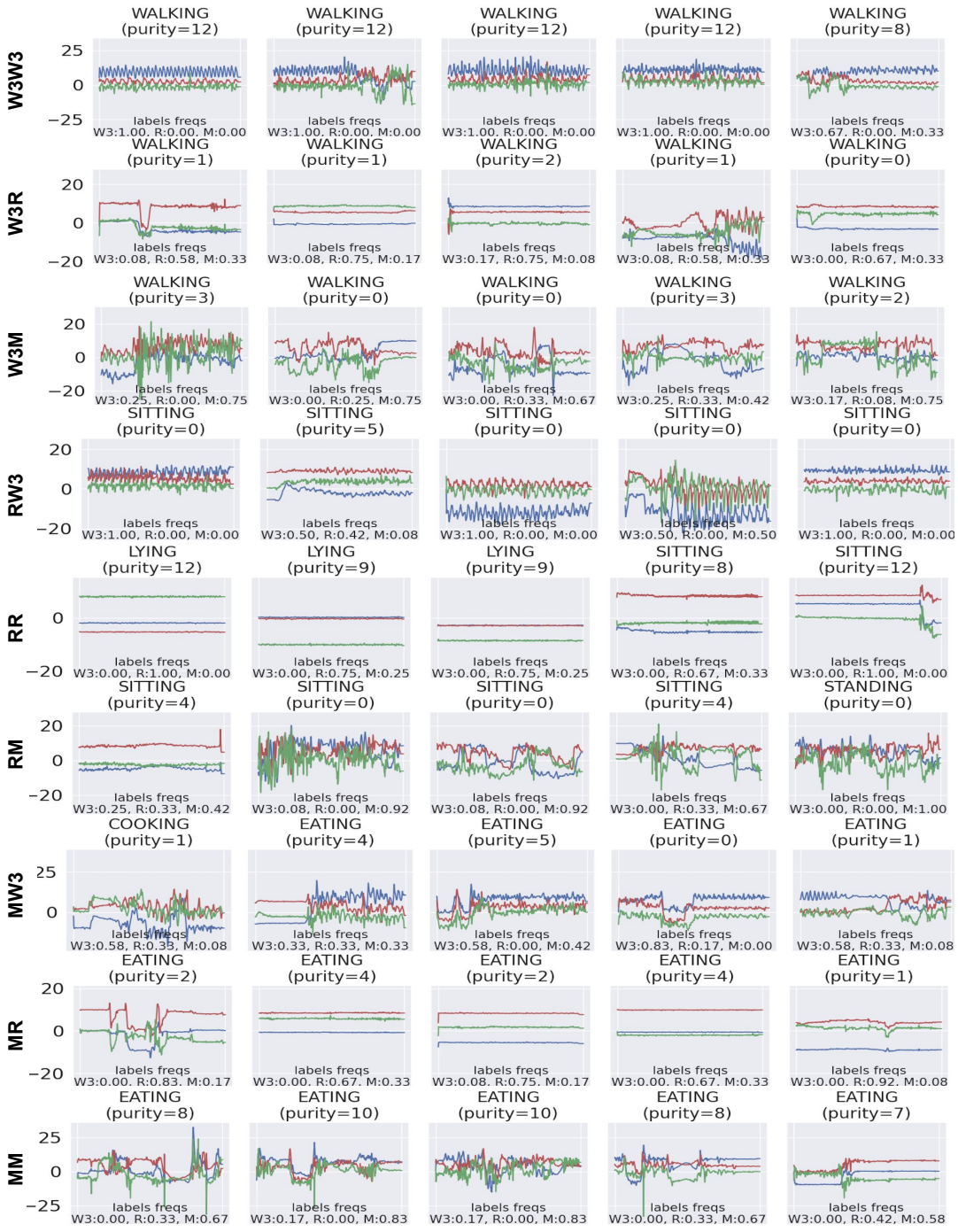


Figure 6.12: Sample visualisation of **rawWatch-ExtraSensory** samples for each true VS estimated label pair in the shared semantic space.

Notation: XY := (true X, estimated Y), where X, Y in {W3 - walk3D; R - relax; M - manual}. For each XY label pair, we visualise 5 random samples from the **ExtraSensory** dataset, including their original label, estimated purity score and frequencies of each label in the output sequence of estimated labels. Interestingly, misclassified samples have low purity scores and exhibit similarities with the samples from their corresponding true positive class (observe, for example, the sample in W3R and RR). This reveals annotation inconsistencies and errors, leading to *label reliability mismatch*.

samples of the class they're classified into or have large segments similar to that class. For example, let us compare samples from the `walk3D` class that are classified as `relax` (W3R) and true positive `relax` samples (RR), respectively. Evidently, the majority of W3R samples are entirely static, much like the true positive RR samples, and are estimated as such with low purity scores. This points to the direction of *label reliability mismatch* due to erroneous label assignments during the dataset's annotation. The remaining W3R samples also have large segments of stationarity, which could be attributed either to erroneous annotations or different perceptions of the same activity by different users. In the following section, we will utilise the newly-defined *purity score* to resolve the *label reliability mismatch*.

6.5.4 Estimation and mitigation of *label reliability mismatch*

In the previous section, we established that **ExtraSensory** dataset suffers from all four levels of mismatch (*label content*, *timescale*, *label definiton* and *label reliability*) with respect to the **watchHAR** dataset. Here, we will aim to identify the subset of the dataset that can be trusted to learn robust models for the **ExtraSensory** data.

6.5.4.1 Target-Domain Mapping

In pursuit of this objective, let us revisit Fig. 6.11 that summarises the purity score distributions for each class of the **ExtraSensory** dataset. We will systematically eliminate the samples with the lowest purity within each class and then train classifiers in each granularity level using the refined subset. This process will be iterated, commencing from purity level 0 and advancing up to purity level 12.

We will examine the following representations for the **ExtraSensory** dataset: the original features released, i.e., the **featWatch-ExtraSensory** dataset, and the learned representations derived from the rawWatch-ExtraSensory dataset when evaluated using the $C_{GL_{N,w}}$ models, namely *Latent*, *Logits*, *Scores*, *labelsSeq* and *labelsFreq*, as these where defined in Section 6.5.2. Using majority voting, we will also report the source model's generalisation performance at the target-timescale.

6.5.4.2 Dataset alignment test #4

In this final mismatch mitigation step, dataset alignment is established by inspecting the source model's generalisation capabilities and target models' learning capacity, as the **ExtraSensory** dataset is gradually cleaned based on its samples' estimated purity scores. Fig. 6.13 summarises the performance of the learned models when varying the purity scores for the **featWatch-ExtraSensory** and all the latent representations of the **rawWatch-ExtraSensory** generated by the source models. The last row summarises the generalisation performance of the source models using the majority voting rule we introduced in 6.5.2.

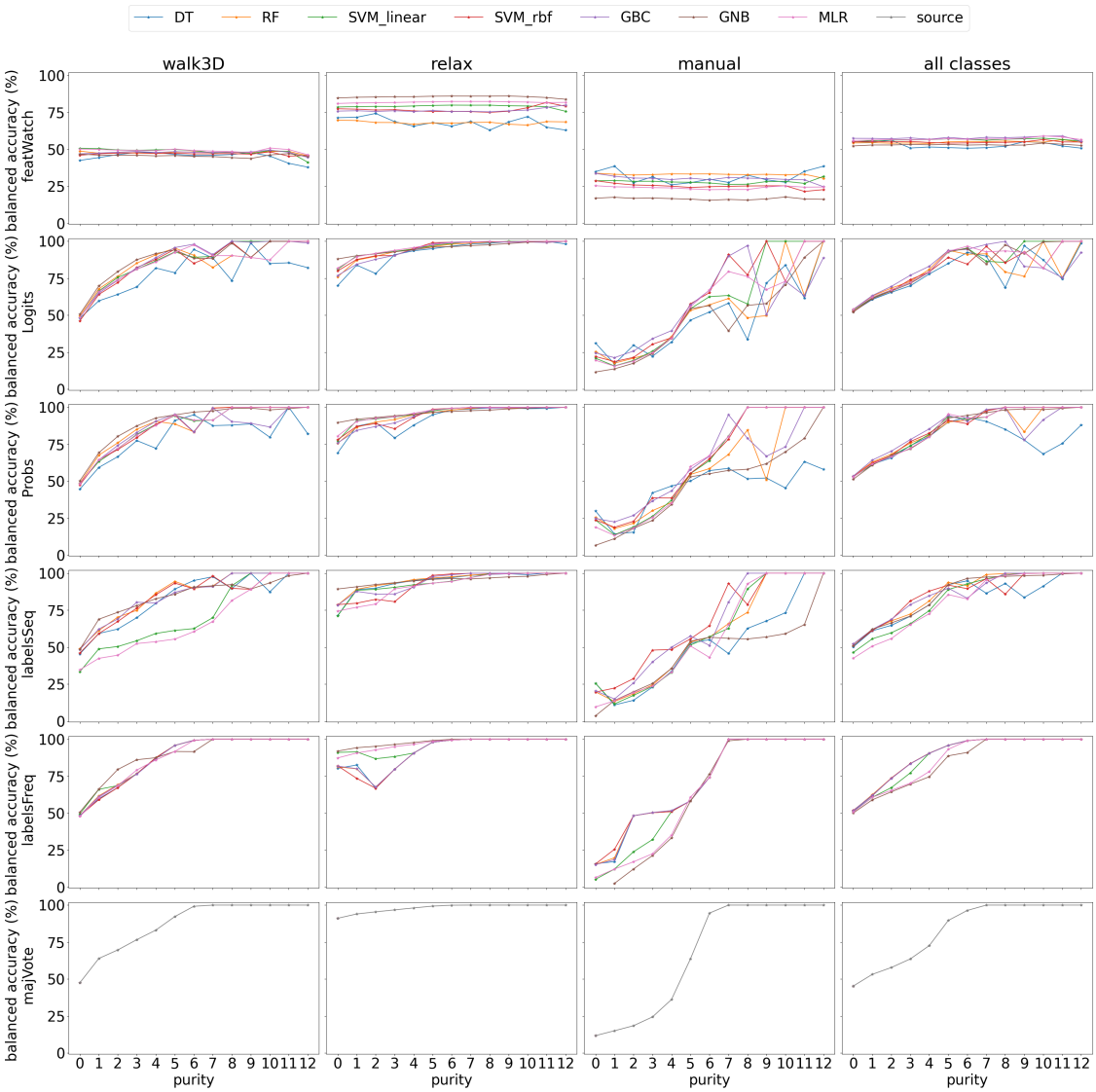


Figure 6.13: Dataset cleaning performance based on purity score for GL_2

Our observations reveal distinct trends: the handcrafted released **featWatch-ExtraSensory** features exhibit challenges in effectively differentiating between classes, even when opting for high-purity samples. In contrast, the acquired latent representations of **rawWatch-ExtraSensory** demonstrate a progressive improvement in constructing more precise models as the selected subset's purity increases, evident in both individual class performance (per class balanced accuracy) and overall model performance (balanced accuracy). While improvement is observed across all latent representations, the *labelsFreq* features are the most robust across all classes and regarding overall performance. Similarly, while most models seem to stabilise towards higher accuracies as the purity of the dataset increases (particularly for $p_{min} \geq 7$, which is the identified purity score establishing robust **cross-domain mapping** across the two datasets), the SVMs (both with linear and rbf kernels) and the multiclass linear regressors seem to be the most robust of models. Finally, the source models effectively generalise on progressively cleaner subsets of the target datasets using the inference-based majority vote approach. The proposed approach of removing data serves as a means of mitigating noise within the dataset. While this method effectively distinguishes between reliable and noisy samples, whether due to "polluted" recordings, where the activity denoted in the recording's annotation is not performed consistently enough to be labelled by the classifier in the same way, or due to erroneous annotations, it also reduces the overallwdataset size. This reduction may introduce challenges such as overfitting and limit opportunities for techniques such as domain adaptation or active learning. In this case, however, we only observe a 25% reduction in training dataset file (30297 target recordings remaining after cleaning, with an original 40241) and 21.5% in testing dataset size (8980 recordings remaining, with an original 11462).

Based on the above, we establish that the *purity score* is a robust mechanism for dataset cleaning when we have another semantically similar dataset to extract semantic information from. It helps resolve the *label reliability mismatch* across datasets, allowing for a trust-based curation of a cleaner, well-defined subset from a real-world, crowdsourced dataset.

Overall, the proposed dataset alignment technique offers the following key contributions:

- It identifies subsets of target data samples that closely resemble those the trained model has learned to recognise (referred to as "good samples" in this comment).
- It provides semantic descriptions of the target data, which highlight the sensitivity of the classifier to "polluted" samples (i.e., samples that do not exhibit the pattern of their given annotation 100% throughout). The calculated purity score for each activity denotes the sensitivity of the classifier for that activity.
- It identifies subsets of the target data that are challenging to classify accurately, even when part of the recording aligns with the given annotation. Additionally, it offers semantic descriptions on what are the other activities occurring within the recording that contribute to classification difficulty.
- It identifies subsets of the erroneously labelled data in the targetand provides semantic descriptions of the actual events taking place during these recordings, which can support the process of re-labelling to enhance data utility.

It is important to note that the purity score is a learnable parameter, computed only during training to capture the patterns associated with each activity in the target domain. At inference time, this score is embedded within the original classifier as a mechanism to inform classification decisions.

Having identified and resolved the *label content*, *timescale*, *label definiton* and *label reliability mismatch* between the **watchHAR** and **ExtraSensory** datasets, we have established label and time alignment between the two datasets. Aligning datasets with semantic similarities in their label and time domains proves valuable, yielding substantial downstream contributions in data augmentation, robust model training and continual learning.

6.6 Related work

Human Activity Recognition models have been extensively studied over the past few decades, achieving high accuracy in recognising a wide range of activities, including indoor, outdoor, and daily living tasks [74, 75, 201–204, 214, 215, 219]. However, like most machine learning models, HAR models are typically trained under the closed-world assumption, which presumes that data at inference time originate from the same distribution as the training data [246]. In real-world settings, this assumption is often violated due to variability in user behaviour, differences in device placement, and sensor heterogeneity [42, 393]. Handling such variations in data distributions is key to real-world model deployment.

Furthermore, with the growing prevalence of mobile and wearable devices, the ability to analyse and interpret crowdsourced data, which are frequently subject to annotation inconsistencies due to their in-the-wild collection [393], has become increasingly important for such datasets to be usable in model training and generalisation. This section examines how various machine learning research domains have addressed these challenges, focusing on generalisation [241, 245, 246, 250, 261, 264, 265, 267, 269, 270, 272, 274, 275], transfer learning [284–287, 290–295, 297], and the impact of noisy annotations [300–304, 306, 307, 313–316, 326, 327] in HAR.

Generalisation The issue of *label content mismatch*, as was defined in this thesis, has found effective solutions in conventional "out-of-distribution (OOD) detection" scenarios, dealing with unidentified classes in incoming samples [240]. Various approaches have been explored, varying from inspecting a classifier's predictions such as ODIN [245], OpenMax [250] and GradNorm [253], to modeling the density of in-distribution (ID) samples with a probabilistic model [262, 264], and measuring the distance of incoming samples from the centroids of known classes [265, 268] or the quality of reconstruction [270, 274]. However, this problem hasn't been extensively explored when incoming samples introduce new concepts but share logical similarities with known classes. Recent research has delved into near-distribution (nearD) detection, focusing on distinguishing samples resembling the

model's training dataset from entirely unrelated data [264, 268]. Yet, this space has not explored mechanisms to identify shared patterns between ID and nearD data; recognising semantic similarities among samples from different but related datasets is crucial for capturing naturalistic human behaviours within HAR models.

Our proposed approach, SeMEDA, builds upon this concept. While not a conventional generalisation method, it addresses a distinct generalisation challenge by providing semantic explanations for incoming data that are semantically coherent with (i.e., have the same annotation space) or lie in the near-distribution space of the training dataset.

Transfer Learning The area of transfer learning offers solutions for a model to adapt in new domains, where classes may be defined differently, resulting in *label content mismatch* [277], by leveraging knowledge from one task (the source domain) to improve learning efficiency and performance in a different but related task (the target domain). Popular methods include TrAdaBoost [278] and GapBoost [282], that re-weight or re-sample (a subset of) instances from the source domain, identifying instances with low scores as coming from a different distribution. In a different direction, parts of a model trained in the source domain may be used in the target domain, such as in notable works DAM [292], MMKD [294] and TransEMDT [295]. Feature-based approaches are also notable, such as CORAL [286], Deep Coral [287], CoCC [290] or the GAN-based DANN [289], transferring or transforming features to a new space, which can be either a new, domain-invariant space or a combination of the source and target domains. However, all these techniques lack explainability in the new domain. At the same time, while adaptations to diverse environments exist, accommodating variations in users, devices, or placements [42, 394, 395], there's a notable absence, to our knowledge, of direct support for predictions at different input sizes (e.g., due to *timescale mismatch*).

While incorporating elements of both feature-based and parameter-based transfer learning, such as leveraging source models as feature extractors for target models,

SeMEDA is not purely a transfer learning problem. Traditional transfer learning aims to adapt trained models for new domains but often lacks insight into why models struggle in these environments. SeMEDA builds upon transfer learning principles to uncover underlying patterns, enhancing interpretability and facilitating more informed mechanisms for improved model adaptability.

Noisy data Mislabelled instances in datasets can disrupt the training process of HAR models, causing them to overfit to the noisy samples and, as a result, impairing both performance and robust generalisation. Research regarding noisy data handling can be categorised as architecture-based, regularisation-based, loss-function-based and sample-selection-based.

Acknowledging potential noise in real-world model deployment annotations leading to *label reliability mismatch*, recent efforts in learning from noisy labels have emerged [392, 396]. These include alterations in model architecture [302], incorporating regularisation mechanisms [300] or appropriate loss functions [314] and dataset cleaning to improve trustworthy dataset curation [327]. However, the estimation of noise type tends to be numerical, lacking a semantic, explainable understanding of incoming data.

SeMEDA is not a classic problem of "learning from noisy data"; while significant efforts in the field have been focused on tackling the *uncertainty due to label noise*, we believe a tangible interpretation of noise in terms of semantic concepts and behaviours is crucial for practical deployment in real-world scenarios. This is a prominent focus of SeMEDA, which borrows from the field of noisy data handling, primarily employing sample-selection mechanisms to identify trustworthy parts of new datasets.

SeMEDA Our proposed approach, SeMEDA, operates at the intersection of the aforementioned research fields, focusing, however, on data understanding from a layperson's perspective, rather than a strictly machine learning-centric viewpoint. It aims to identify both well-documented challenges in crowdsourced datasets, such as noisy or mislabelled samples, and a semantic understanding of incoming activity

data based on user perception. For example, if a user is walking but a mobility app fails to recognise it, it is important to understand how their interpretation differs from what the trained model classifies as walking. While such discrepancies may stem from well-documented challenges in the fields of generalisation and domain adaptation, such as sensor differences or data distribution shifts, they can also arise from subjective annotation inconsistencies across users.

Understanding these semantic variations is key to improving model generalisation and gaining insights into how different users perceive activities. SeMEDA integrates: i) generalisation, by examining the notion of in-distribution and near-distribution semantics across different users and datasets; ii) transfer learning, through techniques such as model/parameter sharing and feature extraction from source models to enhance learning in the target space; and, iii) noisy data handling, incorporating strategies such as dataset cleaning to improve data reliability. By detecting trustworthy dataset segments, flagging mislabelled data, and identifying semantic divergences due to differing user perceptions, SeMEDA enhances model robustness, interpretability and adaptability for real-world deployment.

6.7 Conclusions

In this chapter, we proposed a Semantic Mismatch Estimation and Dataset Alignment approach to identify and resolve the semantic differences between two datasets to ensure the generalisation of existing models and accurate new model training on the new dataset. We considered the specifics of one dataset to be known (source controlled dataset) and the other to be unknown or imprecise (target crowdsourced dataset).

We identified four primary levels of semantic mismatch across such datasets: *label content mismatch*, involving semantically similar but not identical classes; *timescale mismatch*, arising from annotations over different window lengths; *label definition mismatch*, where the same class label is defined with different rules across datasets; and *label reliability mismatch*, stemming from significant variability in label definitions within the target dataset or a notable presence of erroneous labels.

We established a different version of semantic mapping to address each of the above challenges. A *hierarhical label mapping* was introduced to map semantically similar classes across the two datasets over a higher level of semantic abstraction. Then, two approaches to *timescale mapping* were proposed to allow for predictions at different timescales. A *cross-domain mapping* based on the introduced notion of *purity score* enabled the understanding of variabilities in the definition of corresponding abstracted classes across the two datasets. Finally, a systematic *target-domain mapping* approach allowed for the curation of cleaner datasets with trustworthy annotations without the need for laborious data analysis and expert annotation.

We showcased our proposed approach through two datasets: the **watchHAR** dataset, which was collected in controlled laboratory conditions, and the **ExtraSensory** dataset, which was collected in-the-wild. The two selected datasets suffer from all types of mismatches; hence, it is required that all semantic mismatches are resolved for the datasets to be aligned effectively. Through extensive experimental analysis, we revealed the underlying patterns of the **ExtraSensory** dataset through the scope of the **watchHAR** dataset. We demonstrated the potency of learned models in generating robust latent representations by examining the learning capacity of new models using various latent representations generated by the source models. Notably, these models outperformed conventional hand-engineered features, displaying a higher proficiency in capturing and differentiating nuances within the data.

This emphasises the significance of leveraging learned models for real-world applications, particularly in scenarios where datasets present varying semantic nuances. The ability of learned models to discern and reveal semantic complexities in new data showcases their potential in practical deployment, both in terms of real-time inference and in establishing semantic mappings between the incoming data, and the scope of the deployed model.

Such a semantic alignment allows for significant downstream contributions. As we have demonstrated, cleaning the target dataset allows for a higher learning

capacity of models trained on this dataset. At the same time, ensuring consistent annotations across two datasets paves the way for data augmentation by joining the trusted components of the datasets, enabling more accurate training of potentially deeper models. Finally, it is a valuable step towards continual learning by allowing the identification of composite activities as expressions of combinations of known classes (for example, "sitting while ironing") can be identified as a "relaxed-manual task" in our abstracted universe).

Chapter 7

Conclusions

7.1 Concluding remarks

This thesis has investigated seamless, non-invasive, and lightweight solutions to analysing human daily patterns through three key components of human behavioural understanding: location estimation, activity recognition, and task perception.

The widespread availability and social acceptability of smart devices that offer numerous mobile applications to support everyday life have revealed the necessity to implement solutions that satisfy two main characteristics: are non-disruptive to the user's naturalistic behaviour and can support the user in their own private spaces, for example, their home or workplace. As such, they need to be cost-efficient, avoid heavy infrastructure, and ensure the privacy of the user and of others sharing the environment.

To support these requirements, we have proposed in this thesis to address the above problems by utilising only smartwatch inertial sensors, which are lightweight and popular in modern societies, and ambient sensor signals that can be detected in the environment. We have also focused on a semantic interpretation of these tasks, as we believe semantic information is informative enough for behavioural understanding. It also allows for more privacy-preserving solutions for the user,

lifting the requirements for environment floorplans to infer the user's location or for identification of the user's exact pose when performing an activity.

While smartwatch solutions have received limited exploration in the area of ubiquitous computing, in this thesis we demonstrate that the unique properties of smartwatch inertial sensors not only allow for continuous user tracking but also motivate solutions that are friendly to their limited battery power and computing resources compared to other mobile devices, such as smartphones.

Chapter 3.4 introduced two new smartwatch-based datasets, **WatchBLoc** and **watchHAR**, that were collected to support the research conducted in this thesis and contribute to benchmarks in the research community. **WatchBLoc**, in particular, comprises, to our knowledge, the first dataset that offers extensive recordings from ambient anchor nodes and the first to couple ambient sensor information with smartwatch inertial data. **watchHAR** contributes an informative variety of activities of daily living, ensuring the recorded activities are pure and well-annotated.

Chapter 4.7 focused on the indoor positioning problem in privacy-preserving environments. We introduced the complexities and constraints of private indoor spaces and highlighted the suitability of mobile devices, and in particular smartwatches, to support localisation solutions in areas with these unique characteristics. We proposed **RoomE**, a lightweight framework that fuses mobility information from inertial smartwatch data with room occupancy information from ambient sensor nodes, to accurately and continuously track users in private indoor spaces. We systematically evaluated the proposed approach across different users, environments and device placements using the **WatchBLoc** dataset.

Chapter 5.7 explored the problem of Human Activity Recognition for multiple mobile applications concurrently. We established that smart wearable devices, albeit lightweight and capable of continuous monitoring, display limited computational resources and battery power yet are required to support multiple mobile applications at the same time. Based on the observation that many such applications require similar activity information, we proposed a centralised framework that is capable of providing activity information over a hierarchy of decision levels based on

the demands of the mobile applications in action at each point in time. We evaluated the proposed approach on the **watchHAR** dataset, and systematically inspected its inference-time capabilities across three devices of different computing power, highlighting the improvement in computational time compared to standard flat classification.

Chapter 6.7 investigated the inconsistencies in activity data annotations across samples belonging to corresponding semantic classes in different datasets, which may hinder the generalisation capabilities of learned models and inhibit effective knowledge transfer across the two datasets. We systematically identified four levels of such inconsistencies relating to the datasets' semantic content, annotation frequency, task perception, and annotation consistency. To address these challenges, we proposed **SeMEDA**, an inverse-hierarchy approach that borrows from the model generalisation and transfer learning domains to reveal abstracted semantic mappings across HAR datasets and address their identified inconsistencies. We demonstrated our proposed approach on the **watchHAR** and **ExtraSensory** datasets, that contain overlapping and semantically similar classes yet differ in their data collection and annotation mechanisms, being controlled and crowdsourced, respectively. Our experiments highlighted the importance of leveraging learned models to not only transfer knowledge to new datasets but also to understand incoming data from the closed-world perspective of a trained model, paving the way for a variety of downstream tasks, such as dataset cleaning, data augmentation and continual learning.

Overall, our work in this thesis has revealed the potential of smartwatch sensors in providing robust solutions for practical, real-world problems, preserving naturalistic behaviour and without the need for bespoke infrastructure. In the following section, we will explore how our contributed methodologies can further aid related research in ubiquitous computing.

7.2 Future directions

Although in this dissertation we have provided novel directions in human behavioural understanding, there are still underlying limitations and areas for future investigation. In this section, we will discuss the potential research directions enabled by the proposed systems.

7.2.1 Infrastructure-free indoor positioning

In our proposed approach towards semantic localisation of users in private indoor spaces, we have made the assumption that every room in the environment will contain an IoT device that can be sensed by the smartwatch's sensors.

In reality, there might be rooms in the environment where no such device exists, as we do not yet truly live or work in fully smart spaces where all devices are interconnected. In this direction, investigating how a minimal number of IoT devices can provide reliable localisation of the user both for rooms that are equipped with a device and for adjacent rooms that are not, is key to mirroring more realistic living scenarios.

Unsupervised approaches towards continual learning are particularly promising to effectuate such scenarios; by allowing for ambient signals from known room-level anchor points, coupled with mobility information from the smartwatch, to cluster, we should be able to identify ambient signal patterns that signify occupancy of known rooms, or room-changes to new locations of unknown semantics.

Further enhancing the motion estimates to provide not only mobility information but exact activities, can also enable the assignment of semantic labels for newly identified locations; for example, if there exists no ambient sensor in the kitchen, but the system identifies a new location where cooking and washing dishes take place, the model could be extended to identify this new location to correspond to "kitchen" with high probability.

7.2.2 Distributed systems for human activity recognition

Our proposed hierarchical approach to activity recognition was motivated by the concurrent needs of various mobile applications for activity information of a similar nature that may be required at different levels of detail or different timings.

Techniques from the groundbreaking area of natural language processing could be explored in order to group activities based on their conceptual annotation similarities, automating the hierarchy generation and hence removing subjectivity from the system design. Alternatively, employing unsupervised approaches to detect similarities in feature representations of activities could inform the hierarchy design from a signal pattern similarity rather than a semantic perspective.

The set of activities to include in the hierarchy is also an important detail to consider. While in our analysis we have included a combination of "core" and composite activities in the hierarchy, identifying the set of all possible composite tasks at each level of the hierarchy could take a toll on the system's scalability as the number of activities increases. Instead, it would be advisable for our framework to be defined only over a set of simple activities and combine them as and when appropriate. This would require the framework to identify multiple tasks simultaneously in order to identify composite tasks. Research directions towards multi-task and concurrent-task learning are key in this direction.

Another valuable direction is transferring the framework to a distributed setting, where the user is equipped with a multitude of wearable IMU devices, e.g., a smartphone in the pocket, a smartwatch on the wrist, a smart ring on the finger, an in-ear wearable, etc. In this more complicated scenario, we could consider each sensor's computing capabilities and design the framework to optimise resources, by assigning the more computationally demanding branches of the hierarchy to devices with more processing power and longer battery life. Body placement of each device can also affect the hierarchy design. For instance, the existence of a smart ring could further extend the hierarchy to tactile activities, while an in-ear sensor could help identify head movements, e.g., for attention-based tasks.

Finally, a classification model's success stems not only from its predictive accuracy in known tasks but also from its ability to quantify uncertainty when faced with unseen input. Our proposed method operates effectively without calibration, as demonstrated by the Leave-One-Subject-Out evaluation; it is, however, unable to quantify uncertainty, particularly for IMU data from activities it has not been trained to recognise. Such an approach can allow the model to query the user on their current activities when it is highly uncertain of its predictions, allowing both for mini-calibration when required and opening up the way for lifelong learning, extending the classifier's capabilities to new activities as they appear in the user's behavioural patterns.

7.2.3 Predictive uncertainty and continual learning

In our work pertaining to the semantic understanding of new datasets, we have explored a structured approach to handling inconsistencies across semantic classes of two independent datasets when the classes are semantically identical or similar. While in this work we considered datasets where the classes were strictly of this nature, i.e., there existed no classes in the new dataset that were completely unrelated to the known dataset, in the real-world, incoming activities may vary from identical to the ones the model has been trained on (in-distribution), to semantically similar (near-distribution), or completely unrelated (out-of-distribution).

In this direction, ensuring a trained model has uncertainty estimation mechanisms in place is key to identifying not only how activities may be similar to the ones known by the model but also how they might differ. While works in-distribution and out-of-distribution detection have demonstrated significant results in the last few years, the area of near-distribution detection, which is critical for our proposed approach, has received limited consideration. As such, developing methods that can effectively capture the nuances of incoming data that would categorise them as semantically similar to the known classes is key to incorporating our proposed methodology in a real-world context of infinitely many incoming activities. In a supervised setting, when incoming data labels are available, language

models provide a promising direction into identifying data with semantic similarities, automating the identification of similar classes across datasets, as well as informing the inverse-hierarchy structure, as we also suggested for the hierarchical learning framework in the previous section.

7.2.4 Long-term vision

While each of our proposed methods and future directions provides important information towards human behavioural understanding, leveraging the above directions is necessary to provide systems that are informative and reliable in detecting real-world human behaviours.

In this direction, combining positioning and activity estimates can significantly improve robustness; constraining the set of possible activities to the ones that are likely in a particular location and vice-versa can significantly enhance a model's predictive capabilities.

Robust generalisation mechanisms that are capable of identifying whether incoming data are known to, similar to, or completely unrelated from the data a model has been trained on are critical to seamlessly handling the streams of naturalistic human behaviour. Having such measures in place enables the explainability of incoming data from a well-trained model, and opens up important directions to continual learning, allowing for a model to expand its predictive scope when new behavioural patterns become available.

References

[1] Mussab Alaa et al. “A review of smart home applications based on Internet of Things”. In: *Journal of Network and Computer Applications* 97 (2017), pp. 48–65.

[2] Serena Zheng et al. “User Perceptions of Smart Home IoT Privacy”. In: *Proc. ACM Hum.-Comput. Interact.* 2.CSCW (Nov. 2018). URL: https://doi.org/10.1145/3274469.

[3] Jonathan M. Peake, Graham Kerr, and John P. Sullivan. “A Critical Review of Consumer Wearables, Mobile Applications, and Equipment for Providing Biofeedback, Monitoring Stress, and Sleep in Physically Active Populations”. In: *Frontiers in Physiology* 9 (2018).

[4] Lili Liu et al. “Smart homes and home health monitoring technologies for older adults: A systematic review”. In: *International Journal of Medical Informatics* 91 (2016), pp. 44–59.

[5] *Smartphones - statistics & facts.* https://www.statista.com/topics/840/smartphones/#topicOverview. Accessed: 2023-09-08.

[6] *Smartwatches - statistics & facts.* https://www.statista.com/topics/4762/smartwatches/#topicOverview. Accessed: 2023-09-08.

[7] *Wearables - statistics & facts.* https://www.statista.com/topics/1556/wearable-technology/#topicOverview. Accessed: 2023-09-08.

[8] *Wearable Technology Market - Global Industry Analysis, Size, Share, Growth, Trends, Regional Outlook, and Forecast 2023-2032.* https://www.precedenceresearch.com/wearable-technology-market. Accessed: 2023-09-08.

[9] Fabio Sartori. “An API for Wearable Environments Development and Its Application to mHealth Field †”. In: *Sensors* 20.21 (2020). URL: https://www.mdpi.com/1424-8220/20/21/5970.

[10] Mohammad Moshawrab et al. “Smart Wearables for the Detection of Cardiovascular Diseases: A Systematic Literature Review”. In: *Sensors* 23.2 (2023). URL: https://www.mdpi.com/1424-8220/23/2/828.

[11] *Number of mobile devices worldwide 2020-2025*. https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/. Accessed: 2023-09-08.

[12] *Life Gets Better with Mobile Apps, Literally*. https://www.airship.com/blog/life-gets-better-with-mobile-apps-literally/. Accessed: 2023-09-08.

[13] Guochang Xu and Yan Xu. *GPS*. Springer, 2007.

[14] Patrick Dickinson et al. “Indoor positioning of shoppers using a network of Bluetooth Low Energy beacons”. In: *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2016, pp. 1–8.

[15] *Intelligent Location Solution*. https://navenio.com/intelligent-location-solution/. Accessed: 2023-09-08.

[16] Shuo Tian et al. “Smart healthcare: making medical care more intelligent”. In: *Global Health Journal* 3.3 (2019), pp. 62–65.

[17] Giuseppe Aceto, Valerio Persico, and Antonio Pescapé. “Industry 4.0 and Health: Internet of Things, Big Data, and Cloud Computing for Healthcare 4.0”. In: *Journal of Industrial Information Integration* 18 (2020), p. 100129.

[18] Ioana Iancu and Bogdan Iancu. “Designing mobile technology for elderly. A theoretical overview”. In: *Technological Forecasting and Social Change* 155 (2020), p. 119977.

[19] *Why Smart Homes are the Future*. https://www.trustedtechnology.co.uk/tech-news/why-smart-homes-are-the-future. Accessed: 2023-09-08.

[20] *Flat design apps in fitness tracker and man running*. https://www.freepik.com/free-vector/flat-design-apps-fitness-tracker-man-running_7883264.htm. Accessed: 2023-12-21.

[21] *Smartphone with GPS map navigation app*. https://stock.adobe.com/uk/images/smartphone-with-gps-map-navigation-app/227624503. Accessed: 2023-12-21.

[22] *Indoor positioning and navigation using Wi-Fi*. https://navigine.com/blog/wifi-for-indoor-positioning-and-navigation/. Accessed: 2023-12-21.

[23] Deblu Sahu et al. “The Internet of Things in Geriatric Healthcare”. In: *Journal of healthcare engineering* 2021 (July 2021), p. 6611366.

[24] *Benefits Of Smart Home Technology | Infographic*. https://hdsonline.co.uk/benefits-of-smart-home-technology-infographic/. Accessed: 2023-12-21.

[25] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (Oct. 2015), pp. 1147–1163. URL: https://doi.org/10.1109%2Ftro.2015.2463671.

[26] Naser El-Sheimy and You Li. "Indoor navigation: state of the art and future trends". In: *Satellite Navigation* 2.1 (May 2021), p. 7. URL: https://doi.org/10.1186/s43020-021-00041-3.

[27] Toshiaki Koike-Akino et al. "Fingerprinting-Based Indoor Localization With Commercial MMWave WiFi: A Deep Learning Approach". In: *IEEE Access* 8 (2020), pp. 84879–84892.

[28] Damir Arbula and Sandi Ljubic. "Indoor Localization Based on Infrared Angle of Arrival Sensor Network". In: *Sensors* 20.21 (2020). URL: https://www.mdpi.com/1424-8220/20/21/6278.

[29] H. Zou et al. "Standardizing location fingerprints across heterogeneous mobile devices for indoor localization". In: *2016 IEEE Wireless Communications and Networking Conference*. Apr. 2016, pp. 1–6.

[30] Z. Xiao et al. "Robust pedestrian dead reckoning (R-PDR) for arbitrary mobile device placement". In: *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Oct. 2014, pp. 187–196.

[31] Changhao Chen et al. "IONet: Learning to Cure the Curse of Drift in Inertial Odometry". In: *CoRR* abs/1802.02209 (2018). arXiv: 1802.02209. URL: http://arxiv.org/abs/1802.02209.

[32] Roanna Lun and Wenbing Zhao. "A survey of applications and human motion recognition with microsoft kinect". In: *International Journal of Pattern Recognition and Artificial Intelligence* 29.05 (2015), p. 1555008.

[33] Adnan Farooq and Chee Sun Won. "A survey of human action recognition approaches that use an RGB-D sensor". In: *IEIE transactions on smart processing & computing* 4.4 (2015), pp. 281–290.

[34] Yanwen Wang and Yuanqing Zheng. "Modeling RFID Signal Reflection for Contact-Free Activity Recognition". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2.4 (Dec. 2018). URL: https://doi.org/10.1145/3287071.

[35] Akash Deep Singh et al. "RadHAR: Human Activity Recognition from Point Clouds Generated through a Millimeter-Wave Radar". In: *Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems*. mmNets'19. Los Cabos, Mexico: Association for Computing Machinery, 2019, pp. 51–56. URL: https://doi.org/10.1145/3349624.3356768.

[36] Yuheng Wang et al. "m-Activity: Accurate and Real-Time Human Activity Recognition Via Millimeter Wave Radar". In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 8298–8302.

[37] Marcin Straczkiewicz, Peter James, and Jukka-Pekka Onnela. "A systematic review of smartphone-based human activity recognition methods for health research". In: *npj Digital Medicine* 4.1 (2021), p. 148.

[38] B. Almaslukh. "An Effective Deep Autoencoder Approach for Online Smartphone-Based Human Activity Recognition". In: *International Journal of Computer Science and Network Security* 17 (Apr. 2017).

[39] L. Minh Dang et al. "Sensor-based and vision-based human activity recognition: A comprehensive survey". In: *Pattern Recognition* 108 (2020), p. 107561. URL: https://www.sciencedirect.com/science/article/pii/S0031320320303642.

[40] Mehrdad Fazli et al. *HHAR-net: Hierarchical Human Activity Recognition using Neural Networks*. 2020. URL: https://arxiv.org/abs/2010.16052.

[41] Isaac Debache et al. "A Lean and Performant Hierarchical Model for Human Activity Recognition Using Body-Mounted Sensors". In: *Sensors* 20.11 (2020). URL: https://www.mdpi.com/1424-8220/20/11/3090.

[42] Andrea Rosales Sanabria et al. "ContrasGAN: Unsupervised Domain Adaptation in Human Activity Recognition via Adversarial and Contrastive Learning". In: *Pervasive Mob. Comput.* 78.C (Dec. 2021). URL: https://doi.org/10.1016/j.pmcj.2021.101477.

[43] Ryan McConville et al. "A dataset for room level indoor localization using a smart home in a box". In: *Data in Brief* 22 (2019), pp. 1044–1051. URL: https://www.sciencedirect.com/science/article/pii/S2352340919300411.

[44] Daniel Roggen et al. "Collecting complex activity datasets in highly rich networked sensor environments". In: *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. 2010, pp. 233–240.

[45] Attila Reiss and Didier Stricker. "Introducing a New Benchmarked Dataset for Activity Monitoring". In: *2012 16th International Symposium on Wearable Computers*. 2012, pp. 108–109.

[46] Gary M. Weiss, Kenichi Yoneda, and Thaier Hayajneh. "Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living". In: *IEEE Access* 7 (2019), pp. 133190–133202.

[47] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. "Recognizing Detailed Human Context in the Wild from Smartphones and Smartwatches". In: *IEEE Pervasive Computing* 16.4 (2017), pp. 62–74.

[48] Donald B Rubin. "Inference and missing data". In: *Biometrika* 63.3 (1976), pp. 581–592.

[49] Yiran Dong and Chao-Ying Joanne Peng. "Principled missing data methods for researchers". In: *SpringerPlus* 2 (2013), pp. 1–17.

[50] Rajalakshmi Krishnamurthi et al. "An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques". In: *Sensors* 20.21 (2020). URL: https://www.mdpi.com/1424-8220/20/21/6076.

[51] Ming Zhang et al. "IMU Data Processing For Inertial Aided Navigation: A Recurrent Neural Network Based Approach". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 3992–3998.

[52] Zhen Shan et al. "A novel adaptive moving average method for signal denoising in strong noise background". In: *The European Physical Journal Plus* 137.1 (2021), p. 50.

[53] K Berkner and RO Wells. "Wavelet transforms and denoising algorithms". In: *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No. 98CH36284)*. Vol. 2. IEEE. 1998, pp. 1639–1643.

[54] Chang Ho Kang, Sun Young Kim, and Chan Gook Park. "Improvement of a low cost MEMS inertial-GPS integrated system using wavelet denoising techniques". English. In: *International Journal of Aeronautical and Space Sciences* 12.4 (Dec. 2011), pp. 371–378.

[55] Alex G. Quinchia et al. "A Comparison between Different Error Modeling of MEMS Applied to GPS/INS Integrated Systems". In: *Sensors* 13.8 (2013), pp. 9549–9588. URL: https://www.mdpi.com/1424-8220/13/8/9549.

[56] Nikolas Trawny and Stergios I. Roumeliotis. "Indirect Kalman Filter for 3 D Attitude Estimation". In: 2005. URL: https://api.semanticscholar.org/CorpusID:14569549.

[57] John W Graham. "Missing data analysis: Making it work in the real world". In: *Annual review of psychology* 60.1 (2009), pp. 549–576.

[58] Deepak Adhikari et al. "A Comprehensive Survey on Imputation of Missing Data in Internet of Things". In: *ACM Comput. Surv.* 55.7 (Dec. 2022). URL: https://doi.org/10.1145/3533381.

[59] Zengyu Ding et al. "Comparison of Estimating Missing Values in IoT Time Series Data Using Different Interpolation Algorithms". In: *International Journal of Parallel Programming* 48.3 (2020), pp. 534–548.

[60] Nithya N Vijayakumar and Beth Plale. "Missing event prediction in sensor data streams using kalman filters". In: *Knowl. Discov. Sens. Data* 149 (2008), p. 170.

[61] Mahmudul Hasan et al. "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches". In: *Internet of Things* 7 (2019), p. 100059.

[62] Anuroop Gaddam et al. "Detecting sensor faults, anomalies and outliers in the internet of things: A survey on the challenges and solutions". In: *Electronics* 9.3 (2020), p. 511.

[63] Amin Shahraki and Øystein Haugen. "An outlier detection method to improve gathered datasets for network behavior analysis in IoT". In: (2019).

[64] Wai Keng Ngui et al. "Wavelet analysis: mother wavelet selection methods". In: *Applied mechanics and materials* 393 (2013), pp. 953–958.

[65] Isabelle Guyon and Andre Elisseeff. "An introduction to variable and feature selection". In: *J. Mach. Learn. Res.* 3 (2003), pp. 1157–1182. URL: http://portal.acm.org/citation.cfm?id=944919.944968.

[66] Sos Agaian and Yuhuang Zheng. "Human Activity Recognition Based on the Hierarchical Feature Selection and Classification Framework". In: *Journal of Electrical and Computer Engineering* 2015 (2015), p. 140820.

[67] Aiguo Wang et al. "Towards Human Activity Recognition: A Hierarchical Feature Selection Framework." eng. In: *Sensors (Basel)* 18.11 (Oct. 2018).

[68] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[69] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[70] Alex Graves. *Generating Sequences With Recurrent Neural Networks*. 2014. arXiv: 1308.0850 [cs.NE].

[71] Xingjian SHI et al. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015.

[72] Francisco Javier Ordóñez and Daniel Roggen. “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition”. In: *Sensors* 16.1 (2016).

[73] Sakorn Mekruksavanich and Anuchit Jitpattanakul. “LSTM Networks Using Smartphone Data for Sensor-Based Human Activity Recognition in Smart Homes.” In: *Sensors (Basel)* 21.5 (Feb. 2021).

[74] Wenchao Jiang and Zhaozheng Yin. “Human Activity Recognition Using Wearable Sensors by Deep Convolutional Neural Networks”. In: *Proceedings of the 23rd ACM International Conference on Multimedia*. MM ’15. Brisbane, Australia: Association for Computing Machinery, 2015, pp. 1307–1310. URL: https://doi.org/10.1145/2733373.2806333.

[75] Daniele Ravi et al. “Deep learning for human activity recognition: A resource efficient implementation on low-power devices”. In: *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. 2016, pp. 71–76.

[76] M. Mohammadi et al. “Semi-supervised Deep Reinforcement Learning in Support of IoT and Smart City Services”. In: *IEEE Internet of Things Journal* (2017), pp. 1–12.

[77] Joaquín Torres-Sospedra et al. “UJIIndoorLoc-Mag: A new database for magnetic field-based localization problems”. In: *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2015, pp. 1–10.

[78] Judit Tamás Zsolt Tóth. “Miskolc IIS Hybrid IPS: Dataset for Hybrid Indoor Positioning”. In: *26st International Conference on Radioelektronika*. IEEE. 2016, pp. 408–412.

[79] Jorge Reyes-Ortiz et al. *Human Activity Recognition Using Smartphones*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C54S4K. 2012.

[80] Yonatan Vaizman, Nadir Weibel, and Gert Lanckriet. “Context Recognition In-the-Wild: Unified Model for Multi-Modal Sensors and Multi-Label Classification”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1.4 (Jan. 2018). URL: https://doi.org/10.1145/3161192.

[81] Jinwang Yi et al. “Joint Time Synchronization and Tracking for Mobile Underwater Systems”. In: *Proceedings of the Eighth ACM International Conference on Underwater Networks and Systems*. WUWNet ’13. Kaohsiung, Taiwan: ACM, 2013, 38:1–38:8. URL: http://doi.acm.org/10.1145/2532378.2532404.

[82] Xiuzhen Cheng et al. “Silent positioning in underwater acoustic sensor networks”. English (US). In: *IEEE Transactions on Vehicular Technology* 57.3 (May 2008), pp. 1756–1766.

[83] Rong Peng and Mihail Sichitiu. “Angle of Arrival Localization for Wireless Sensor Networks”. In: 1 (Oct. 2006), pp. 374–382.

[84] Antoni Perez-Navarro et al. “1 - Challenges of Fingerprinting in Indoor Positioning and Navigation”. In: *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*. Ed. by Jordi Conesa et al. Intelligent Data-Centric Systems. Academic Press, 2019, pp. 1–20. URL: http://www.sciencedirect.com/science/article/pii/B9780128131893000010.

[85] “Global Navigation Satellite System Data Errors”. In: *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, Ltd, 2007. Chap. 5, pp. 144–198. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470099728.ch5. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470099728.ch5.

[86] Robert Harle. “A Survey of Indoor Inertial Positioning Systems for Pedestrians”. In: *IEEE Communications Surveys & Tutorials* 15.3 (2013), pp. 1281–1293.

[87] Richard Szeliski. *Computer vision algorithms and applications*. 2011.

[88] Antonio Brunetti et al. “Computer vision and deep learning techniques for pedestrian detection and tracking: A survey”. In: *Neurocomputing* 300 (2018), pp. 17–33.

[89] Alberto J. Palma et al. “Evolution of Indoor Positioning Technologies: A Survey”. In: *Journal of Sensors* 2017 (2017), p. 2630413.

[90] Tasnia Ashrafi Heya et al. “Image processing based indoor localization system for assisting visually impaired people”. In: *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*. IEEE. 2018, pp. 1–7.

[91] Kabalan Chaccour and Georges Badr. “Computer vision guidance system for indoor navigation of visually impaired people”. In: *2016 IEEE 8th international conference on intelligent systems (IS)*. IEEE. 2016, pp. 449–454.

[92] Jae-Hong Shim and Young-Im Cho. “A mobile robot localization using external surveillance cameras at indoor”. In: *Procedia Computer Science* 56 (2015), pp. 502–507.

[93] Yongliang Sun et al. “Device-free human localization using panoramic camera and indoor map”. In: *2016 IEEE International conference on consumer electronics-China (ICCE-China)*. IEEE. 2016, pp. 1–5.

[94] Ákos Utasi and Csaba Benedek. “A Bayesian approach on people localization in multicamera systems”. In: *IEEE transactions on circuits and systems for video technology* 23.1 (2012), pp. 105–115.

[95] Adrian Cosma, Ion Emilian Radoi, and Valentin Radu. “Camloc: Pedestrian location estimation through body pose estimation on smart cameras”. In: *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE. 2019, pp. 1–8.

[96] Min Sun et al. “See-your-room: Indoor localization with camera vision”. In: *Proceedings of the ACM turing celebration conference-China*. 2019, pp. 1–5.

[97] Mohit Jain et al. “Mobiceil: cost-free indoor localizer for office buildings”. In: *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 2018, pp. 1–6.

[98] Bernd Kitt, Andreas Geiger, and Henning Lategahn. “Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme”. In: *2010 IEEE Intelligent Vehicles Symposium*. 2010, pp. 486–492.

[99] D. Nister, O. Naroditsky, and J. Bergen. “Visual odometry”. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 1. 2004, pp. I–I.

[100] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. “DTAM: Dense tracking and mapping in real-time”. In: *2011 International Conference on Computer Vision*. 2011, pp. 2320–2327.

[101] Seok-Ju Lee et al. “QR-code based Localization for Indoor Mobile Robot with validation using a 3D optical tracking instrument”. In: *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 2015, pp. 965–970.

[102] Peter Lightbody, Tomas Krajnik, and Marc Hanheide. “An efficient visual fiducial localisation system”. In: *ACM SIGAPP Applied Computing Review* 17.3 (2017), pp. 28–37.

[103] Feng Hu, Zhigang Zhu, and Jianting Zhang. “Mobile panoramic vision for assisting the blind via indexing and localization”. In: *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III 13*. Springer. 2015, pp. 600–614.

[104] Yicheng Bai et al. “Landmark-based indoor positioning for visually impaired individuals”. In: *2014 12th International Conference on Signal Processing (ICSP)*. IEEE. 2014, pp. 668–671.

[105] Aoran Xiao et al. “An indoor positioning system based on static objects in large indoor scenes by using smartphone cameras”. In: *Sensors* 18.7 (2018), p. 2229.

[106] Jakob J. Engel, Jürgen Sturm, and Daniel Cremers. “Semi-dense Visual Odometry for a Monocular Camera”. In: *2013 IEEE International Conference on Computer Vision* (2013), pp. 1449–1456. URL: `https://api.semanticscholar.org/CorpusID:7110290`.

[107] Huizhong Zhou et al. “StructSLAM: Visual SLAM with building structure lines”. In: *IEEE Transactions on Vehicular Technology* 64.4 (2015), pp. 1364–1375.

[108] Linhui Xiao et al. “Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment”. In: *Robotics and Autonomous Systems* 117 (2019), pp. 1–16.

[109] Alessandro Mulloni, Hartmut Seichter, and Dieter Schmalstieg. "Handheld Augmented Reality Indoor Navigation with Activity-Based Instructions". In: *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. MobileHCI '11. Stockholm, Sweden: Association for Computing Machinery, 2011, pp. 211–220. URL: https://doi.org/10.1145/2037373.2037406.

[110] Andreas Möller et al. "A Mobile Indoor Navigation System Interface Adapted to Vision-Based Localization". In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*. MUM '12. Ulm, Germany: Association for Computing Machinery, 2012. URL: https://doi.org/10.1145/2406367.2406372.

[111] Georg Gerstweiler. "Guiding people in complex indoor environments using augmented reality". In: *2018 IEEE conference on virtual reality and 3D user interfaces (VR)*. IEEE. 2018, pp. 801–802.

[112] Francis Baek, Inhae Ha, and Hyoungkwan Kim. "Augmented reality system for facility management using image-based indoor localization". In: *Automation in construction* 99 (2019), pp. 18–26.

[113] Andy Ward, Alan Jones, and Andy Hopper. "A new location technique for the active office". In: *IEEE Personal communications* 4.5 (1997), pp. 42–47.

[114] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. "The cricket location-support system". In: *Proceedings of the 6th annual international conference on Mobile computing and networking*. 2000, pp. 32–43.

[115] Karalikkadan Ashhar et al. "A narrowband ultrasonic ranging method for multiple moving sensor nodes". In: *IEEE sensors journal* 19.15 (2019), pp. 6289–6297.

[116] A. Mandal et al. "Beep: 3D indoor positioning using audible sound". In: *Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005*. 2005, pp. 348–353.

[117] Chunyi Peng, Guobin Shen, and Yongguang Zhang. "BeepBeep: A High-Accuracy Acoustic-Based System for Ranging and Localization Using COTS Devices". In: *ACM Trans. Embed. Comput. Syst.* 11.1 (Apr. 2012). URL: https://doi.org/10.1145/2146417.2146421.

[118] Kaikai Liu, Xinxin Liu, and Xiaolin Li. "Guoguo: Enabling Fine-Grained Indoor Localization via Smartphone". In: *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys '13. Taipei, Taiwan: Association for Computing Machinery, 2013, pp. 235–248. URL: https://doi.org/10.1145/2462456.2464450.

[119] Ernesto Martín Gorostiza et al. "Infrared Sensor System for Mobile-Robot Positioning in Intelligent Spaces". In: *Sensors* 11.5 (2011), pp. 5416–5438.

[120] Roy Want et al. "The Active Badge Location System". In: *ACM Trans. Inf. Syst.* 10.1 (Jan. 1992), pp. 91–102. URL: https://doi.org/10.1145/128756.128759.

[121] Wilson Sakpere, Michael Adeyeye-Oshin, and Nhlanhla B.W. Mlitwa. "A state-of-the-art survey of indoor positioning and navigation systems and technologies". In: *South African Computer Journal* 29 (May 2017), pp. 145–197. URL: http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S2313-78352017000300009&nrm=iso.

[122] Chao-Chung Peng, Yun-Ting Wang, and Chieh-Li Chen. "LIDAR based scan matching for indoor localization". In: *2017 IEEE/SICE International Symposium on System Integration (SII)*. 2017, pp. 139–144.

[123] Federico Boniardi et al. "Robust LiDAR-based localization in architectural floor plans". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 3318–3324.

[124] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. "A Survey of Indoor Localization Systems and Technologies". In: *IEEE Communications Surveys & Tutorials* 21.3 (2019), pp. 2568–2599.

[125] Toshihiko Komine and Masao Nakagawa. "Fundamental analysis for visible-light communication system using LED lights". In: *IEEE transactions on Consumer Electronics* 50.1 (2004), pp. 100–107.

[126] Weizhi Zhang, MI Sakib Chowdhury, and Mohsen Kavehrad. "Asynchronous indoor positioning system based on visible light communications". In: *Optical Engineering* 53.4 (2014), pp. 045105–045105.

[127] Aaron Ganick and Daniel Ryan. *Method and system for modulating a light source in a light based positioning system using a DC bias*. US Patent 8,334,901. Dec. 2012.

[128] S. Gezici et al. "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks". In: *IEEE Signal Processing Magazine* 22.4 (2005), pp. 70–84.

[129] Fazeelat Mazhar, Muhammad Gufran Khan, and Benny Sällberg. "Precise Indoor Positioning Using UWB: A Review of Methods, Algorithms and Implementations". In: *Wireless Personal Communications* 97.3 (2017), pp. 4467–4491.

[130] Yanying Gu, Anthony Lo, and Ignas Niemegeers. "A survey of indoor positioning systems for wireless personal networks". In: *IEEE Communications Surveys & Tutorials* 11.1 (2009), pp. 13–32.

[131] Gabriel Deak, Kevin Curran, and Joan Condell. "A survey of active and passive indoor localisation systems". In: *Computer Communications* 35.16 (2012), pp. 1939–1954. URL: https://www.sciencedirect.com/science/article/pii/S014036641200196X.

[132] Bai Yan and Lu Xiaochun. "Research on UWB indoor positioning based on TDOA technique". In: *2009 9th International Conference on Electronic Measurement & Instruments*. IEEE. 2009, pp. 1–167.

[133] Pete Steggles and Stephan Gschwind. "The Ubisense smart space platform". In: (2005).

[134] Hui Liu et al. “Survey of Wireless Indoor Positioning Techniques and Systems”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (2007), pp. 1067–1080.

[135] Sergio F. Ochoa et al. “Crowdsource Based Indoor Localization by Uncalibrated Heterogeneous Wi-Fi Devices”. In: *Mobile Information Systems* 2016 (2016), p. 4916563.

[136] P. Bahl and V.N. Padmanabhan. “RADAR: an in-building RF-based user location and tracking system”. In: *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*. Vol. 2. 2000, 775–784 vol.2.

[137] B. Wang et al. “Indoor Localization Based on Curve Fitting and Location Search Using Received Signal Strength”. In: *IEEE Transactions on Industrial Electronics* 62.1 (Jan. 2015), pp. 572–582.

[138] K. Lin et al. “Enhanced Fingerprinting and Trajectory Prediction for IoT Localization in Smart Buildings”. In: *IEEE Transactions on Automation Science and Engineering* 13.3 (July 2016), pp. 1294–1307.

[139] X. Zhao et al. “Does BTLE measure up against WiFi? A comparison of indoor location performance”. In: *European Wireless 2014; 20th European Wireless Conference*. Mar. 2014, pp. 1–6.

[140] Ramsey Faragher and Robert K. Harle. “An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications”. In: 2014.

[141] Charu Jain et al. “Low-cost BLE based Indoor Localization using RSSI Fingerprinting and Machine Learning”. In: *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. 2021, pp. 363–367.

[142] Thomas Tegou et al. “A low-cost room-level indoor localization system with easy setup for medical applications”. In: *2018 11th IFIP Wireless and Mobile Networking Conference (WMNC)*. 2018, pp. 1–7.

[143] Imad Afyouni et al. “Passive BLE Sensing for Indoor Pattern Recognition and Tracking”. In: *Procedia Computer Science* 191 (2021). The 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), The 16th International Conference on Future Networks and Communications (FNC), The 11th International Conference on Sustainable Energy Information Technology, pp. 223–229. URL: https://www.sciencedirect.com/science/article/pii/S187705092101423X.

[144] Lu Bai et al. “A Low Cost Indoor Positioning System Using Bluetooth Low Energy”. In: *IEEE Access* 8 (2020), pp. 136858–136871.

[145] Samer S. Saab and Hamze Msheik. “Novel RFID-Based Pose Estimation Using Single Stationary Antenna”. In: *IEEE Transactions on Industrial Electronics* 63.3 (2016), pp. 1842–1852.

[146] Navid Fallah et al. “Indoor Human Navigation Systems: A Survey”. In: *Interacting with Computers* 25.1 (Jan. 2013), pp. 21–33. eprint: https://academic.oup.com/iwc/article-pdf/25/1/21/2323898/iws010.pdf. URL: https://doi.org/10.1093/iwc/iws010.

[147] L.M. Ni et al. “LANDMARC: indoor location sensing using active RFID”. In: *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).* 2003, pp. 407–415.

[148] C. Wang, H. Wu, and N.-F. Tzeng. “RFID-Based 3-D Positioning Schemes”. In: *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*. 2007, pp. 1235–1243.

[149] Daqiang Zhang et al. “Real-Time Locating Systems Using Active RFID for Internet of Things”. In: *IEEE Systems Journal* 10.3 (2016), pp. 1226–1235.

[150] Abdelmoula Bekkali, Horacio Sanson, and Mitsuji Matsumoto. “RFID Indoor Positioning Based on Probabilistic RFID Map and Kalman Filtering”. In: *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007)*. 2007, pp. 21–21.

[151] Frédéric Bergeron et al. “Indoor Positioning System for Smart Homes Based on Decision Trees and Passive RFID”. In: *Advances in Knowledge Discovery and Data Mining*. Ed. by James Bailey et al. Cham: Springer International Publishing, 2016, pp. 42–53.

[152] Zheng Yang et al. “Mobility Increases Localizability: A Survey on Wireless Indoor Localization Using Inertial Sensors”. In: *ACM Comput. Surv.* 47.3 (Apr. 2015), 54:1–54:34. URL: http://doi.acm.org/10.1145/2676430.

[153] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. “Analysis and Modeling of Inertial Sensors Using Allan Variance”. In: *IEEE Transactions on Instrumentation and Measurement* 57 (2008), pp. 140–149. URL: https://api.semanticscholar.org/CorpusID:16418752.

[154] Soh Khim Ong, ML Yuan, and Andrew YC Nee. “Augmented reality applications in manufacturing: a survey”. In: *International journal of production research* 46.10 (2008), pp. 2707–2742.

[155] Xiaoping Yun et al. “Self-contained position tracking of human movement using small inertial/magnetic sensor modules”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE. 2007, pp. 2526–2533.

[156] Manon Kok, Jeroen D Hol, and Thomas B Schön. “Using inertial sensors for position and orientation estimation”. In: *arXiv preprint arXiv:1704.06053* (2017).

[157] Alejandro Correa et al. “A review of pedestrian indoor positioning systems for mass market applications”. In: *Sensors* 17.8 (2017), p. 1927.

[158] Ada S Cheung et al. “Androgen deprivation causes selective deficits in the biomechanical leg muscle function of men during walking: a prospective case–control study”. In: *Journal of cachexia, sarcopenia and muscle* 8.1 (2017), pp. 102–112.

[159] Wenchao Zhang et al. “A foot-mounted pdr system based on imu/ekf+ hmm+ zupt+ zaru+ hdr+ compass algorithm”. In: *2017 International conference on indoor positioning and indoor navigation (IPIN)*. IEEE. 2017, pp. 1–5.

[160] Mingyang Zhang et al. “Indoor localization fusing wifi with smartphone inertial sensors using lstm networks”. In: *IEEE Internet of Things Journal* 8.17 (2021), pp. 13608–13623.

[161] Jianfan Chen et al. “A data-driven inertial navigation/Bluetooth fusion algorithm for indoor localization”. In: *IEEE Sensors Journal* 22.6 (2021), pp. 5288–5301.

[162] Xuechen Chen, Shupeng Song, and Jihong Xing. “A ToA/IMU indoor positioning system by extended Kalman filter, particle filter and MAP algorithms”. In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2016, pp. 1–7.

[163] E. Paperno, I. Sasada, and E. Leonovich. “A new method for magnetic position and orientation tracking”. In: *IEEE Transactions on Magnetics* 37.4 (2001), pp. 1938–1940.

[164] Joerg Blankenbach, Abdelmoumen Norrdine, and Hendrik Hellmers. “A robust and precise 3D indoor positioning system for harsh environments”. In: *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2012, pp. 1–8.

[165] Seong-Eun Kim et al. “Indoor positioning system using geomagnetic anomalies for smartphones”. In: *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2012, pp. 1–5.

[166] Brandon Gozick et al. “Magnetic Maps for Indoor Navigation”. In: *IEEE Transactions on Instrumentation and Measurement* 60.12 (2011), pp. 3883–3891.

[167] Shuang Song et al. “Real time algorithm for magnet’s localization in capsule endoscope”. In: *2009 IEEE International Conference on Automation and Logistics*. 2009, pp. 2030–2035.

[168] William Storms, Jeremiah Shockley, and John Raquet. “Magnetic field navigation in an indoor environment”. In: *2010 Ubiquitous Positioning Indoor Navigation and Location Based Service*. 2010, pp. 1–10.

[169] Jaewoo Chung et al. “Indoor Location Sensing Using Geo-Magnetism”. In: MobiSys ’11. Bethesda, Maryland, USA: Association for Computing Machinery, 2011, pp. 141–154. URL: https://doi.org/10.1145/1999995.2000010.

[170] Binghao Li et al. “How feasible is the use of magnetic field alone for indoor positioning?” In: *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2012, pp. 1–9.

[171] Stefano Rosa et al. “Semantic Place Understanding for Human–Robot Coexistence—Toward Intelligent Workplaces”. In: *IEEE Transactions on Human-Machine Systems* 49.2 (2019), pp. 160–170.

[172] Michael Hardegger et al. “S-SMART: A Unified Bayesian Framework for Simultaneous Semantic Mapping, Activity Recognition, and Tracking”. In: *ACM Trans. Intell. Syst. Technol.* 7.3 (Feb. 2016). URL: https://doi.org/10.1145/2824286.

[173] Xiaoguang Niu et al. “AtLAS: An Activity-Based Indoor Localization and Semantic Labeling Mechanism for Residences”. In: *IEEE Internet of Things Journal* 7.10 (2020), pp. 10606–10622.

[174] Athina Tsanousa et al. “Combining RSSI and Accelerometer Features for Room-Level Localization”. In: *Sensors* 21.8 (2021). URL: https://www.mdpi.com/1424-8220/21/8/2723.

[175] Baoding Zhou et al. “A Robust Crowdsourcing-Based Indoor Localization System”. en. In: *Sensors (Basel)* 17.4 (Apr. 2017).

[176] Sreenivasan Ramasamy Ramamurthy and Nirmalya Roy. “Recent trends in machine learning for human activity recognition-A survey”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8 (Mar. 2018), e1254.

[177] Kaixuan Chen et al. “Deep Learning for Sensor-Based Human Activity Recognition: Overview, Challenges, and Opportunities”. In: *ACM Comput. Surv.* 54.4 (May 2021). URL: https://doi.org/10.1145/3447744.

[178] Jindong Wang et al. “Deep Transfer Learning for Cross-Domain Activity Recognition”. In: *Proceedings of the 3rd International Conference on Crowd Science and Engineering*. ICCSE’18. Singapore, Singapore: Association for Computing Machinery, 2018. URL: https://doi.org/10.1145/3265689.3265705.

[179] H M Sajjad Hossain, Md Abdullah Al Hafiz Khan, and Nirmalya Roy. “Active learning enabled activity recognition”. In: *Pervasive and Mobile Computing* 38 (2017), pp. 312–330.

[180] Junyi Ma et al. “A survey on wi-fi based contactless activity recognition”. In: *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. IEEE. 2016, pp. 1086–1091.

[181] Diane Cook, Kyle D. Feuz, and Narayanan C. Krishnan. “Transfer learning for activity recognition: a survey”. In: *Knowledge and Information Systems* 36.3 (2013), pp. 537–556.

[182] Muhammad Muaaz et al. “Wi-Sense: a passive human activity recognition system using Wi-Fi and convolutional neural network and its integration in health information systems”. In: *Annals of Telecommunications* 77.3 (2022), pp. 163–175.

[183] Daniel Weinland, Remi Ronfard, and Edmond Boyer. “A survey of vision-based methods for action representation, segmentation and recognition”. In: *Computer Vision and Image Understanding* 115.2 (2011), pp. 224–241. URL: https://www.sciencedirect.com/science/article/pii/S1077314210002171.

[184] Oluwatoyin P. Popoola and Kejun Wang. “Video-Based Abnormal Human Behavior Recognition—A Review”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.6 (2012), pp. 865–878.

[185] Ahmad Jalal, Shaharyar Kamal, and Daijin Kim. “A Depth Video Sensor-Based Life-Logging Human Activity Recognition System for Elderly Care in Smart Indoor Environments”. In: *Sensors* 14.7 (2014), pp. 11735–11759. URL: https://www.mdpi.com/1424-8220/14/7/11735.

[186] Michael Holte et al. “Human Pose Estimation and Activity Recognition From Multi-View Videos: Comparative Explorations of Recent Developments”. In: *Selected Topics in Signal Processing, IEEE Journal of* 6 (Sept. 2012), pp. 538–552.

[187] Zafar Ali Khan and Won Sohn. "Abnormal human activity recognition system based on R-transform and kernel discriminant technique for elderly home care". In: *IEEE Transactions on Consumer Electronics* 57 (2011). URL: https://api.semanticscholar.org/CorpusID:25319150.

[188] Md. Zia Uddin et al. "Human Activity Recognition Using Body Joint-Angle Features and Hidden Markov Model". In: *ETRI Journal* 33 (Aug. 2011).

[189] Hong Cheng et al. "Real World Activity Summary for Senior Home Monitoring". In: *Multimedia Tools Appl.* 70.1 (May 2014), pp. 177–197. URL: https://doi.org/10.1007/s11042-012-1162-5.

[190] Rim Romdhane, Carlos Crispim, and Monique Thonnat. "Activity Recognition and Uncertain Knowledge in Video Scenes". In: Aug. 2013, pp. 377–382.

[191] S. Helal et al. "The Gator Tech Smart House: a programmable pervasive space". In: *Computer* 38.3 (2005), pp. 50–60.

[192] Gregory D. Abowd et al. "The Aware Home: A living laboratory for technologies for successful aging". In: 2002. URL: https://api.semanticscholar.org/CorpusID:11330087.

[193] M. Philipose et al. "Inferring activities from interactions with objects". In: *IEEE Pervasive Computing* 3.4 (2004), pp. 50–57.

[194] K.P. Fishkin, M. Philipose, and A. Rea. "Hands-on RFID: Wireless wearables for detecting use of objects". In: vol. 2005. Nov. 2005, pp. 38–41.

[195] Michael Buettner et al. "Recognizing Daily Activities with RFID-Based Sensors". In: *Proceedings of the 11th International Conference on Ubiquitous Computing*. UbiComp '09. Orlando, Florida, USA: Association for Computing Machinery, 2009, pp. 51–60. URL: https://doi.org/10.1145/1620545.1620553.

[196] Tim van Kasteren et al. "Accurate Activity Recognition in a Home Setting". In: *Proceedings of the 10th International Conference on Ubiquitous Computing*. UbiComp '08. Seoul, Korea: Association for Computing Machinery, 2008, pp. 1–9. URL: https://doi.org/10.1145/1409635.1409637.

[197] D. Cook et al. "Collecting and disseminating smart home sensor data in the CASAS project". In: *Proc. of CHI09 Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*. 2009.

[198] Geetika Singla, Diane J Cook, and Maureen Schmitter-Edgecombe. "Recognizing independent and joint activities among multiple residents in smart environments". en. In: *J Ambient Intell Humaniz Comput* 1.1 (Mar. 2010), pp. 57–63.

[199] Fco. Javier Ordóñez, Paula De Toledo, and Araceli Sanchis. "Activity Recognition Using Hybrid Generative/Discriminative Models on Home Environments Using Binary Sensors". In: *Sensors* 13.5 (2013), pp. 5460–5477. URL: https://www.mdpi.com/1424-8220/13/5/5460.

[200] Iram Fatima et al. "A Unified Framework for Activity Recognition-Based Behavior Analysis and Action Prediction in Smart Homes". In: *Sensors* 13.2 (2013), pp. 2682–2699. URL: https://www.mdpi.com/1424-8220/13/2/2682.

[201] Nishkam Ravi et al. "Activity recognition from accelerometer data". In: *In Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence(IAAI*. AAAI Press, 2005, pp. 1541–1546.

[202] Andreas Bulling, Ulf Blanke, and Bernt Schiele. “A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors”. In: *ACM Comput. Surv.* 46.3 (Jan. 2014). URL: https://doi.org/10.1145/2499621.

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

[204] Thomas Plötz, Nils Y. Hammerla, and Patrick Olivier. “Feature Learning for Activity Recognition in Ubiquitous Computing”. In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*. IJCAI’11. Barcelona, Catalonia, Spain: AAAI Press, 2011, pp. 1729–1734.

[205] Muhammad Hameed Siddiqi, Adil Mehmood Khan, et al. “Video Based Human Activity Recognition using Wavelet Transform and Hidden Conditional Random Fields (HCRF)”. In: *International Conference on Computer Vision Theory and Applications*. Vol. 2. SCITEPRESS. 2012, pp. 401–404.

[206] Manish Khare and Moongu Jeon. “Multi-resolution approach to human activity recognition in video sequence based on combination of complex wavelet transform, Local Binary Pattern and Zernike moment”. In: *Multimedia Tools and Applications* (2022), pp. 1–30.

[207] Alihossein Aryanfar et al. “Multi-view human action recognition using wavelet data reduction and multi-class classification”. In: *Procedia Computer Science* 62 (2015), pp. 585–592.

[208] DK Vishwakarma, Prachi Rawat, and Rajiv Kapoor. “Human activity recognition using gabor wavelet transform and ridgelet transform”. In: *Procedia Computer Science* 57 (2015), pp. 630–636.

[209] Abbas Shah Syed et al. “A hierarchical approach to activity recognition and fall detection using wavelets and adaptive pooling”. In: *Sensors* 21.19 (2021), p. 6653.

[210] Thurmon E Lockhart et al. “Wavelet based automated postural event detection and activity classification with single IMU (TEMPO)”. In: *Biomedical sciences instrumentation* 49 (2013), p. 224.

[211] Yiming Tian et al. “Robust human activity recognition using single accelerometer via wavelet energy spectrum features and ensemble feature selection”. In: *Systems Science & Control Engineering* 8.1 (2020), pp. 83–96.

[212] Anna Nedorubova, Alena Kadyrova, and Aleksey Khlyupin. “Human Activity Recognition using Continuous Wavelet Transform and Convolutional Neural Networks”. In: *arXiv preprint arXiv:2106.12666* (2021).

[213] Thurdsak Leauhatong, Nitipat Nuttaitanakul, and Benjawan Prapochanang. “Optimal selection of mother wavelet for classifying human activities from acceleration signals”. In: *2015 8th Biomedical Engineering International Conference (BMEiCON)*. 2015, pp. 1–5.

[214] Vincent Rahn et al. “Optimal Sensor Placement for Human Activity Recognition with a Minimal Smartphone–IMU Setup”. In: *Proceedings of the 10th International Conference on Sensor Networks - Volume 1: SENSORNETS,* INSTICC. SciTePress, 2021, pp. 37–48.

[215] Isah A. Lawal and Sophia Bano. "Deep Human Activity Recognition With Localisation of Wearable Sensors". In: *IEEE Access* 8 (2020), pp. 155060–155070.

[216] Fuqiang Gu et al. "Locomotion Activity Recognition Using Stacked Denoising Autoencoders". In: *IEEE Internet of Things Journal* 5 (Apr. 2018), pp. 2085–2093.

[217] Shuochao Yao et al. "DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing". In: *Proceedings of the 26th International Conference on World Wide Web*. WWW '17. Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, pp. 351–360. URL: https://doi.org/10.1145/3038912.3052577.

[218] Haodong Guo et al. "Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (2016). URL: https://api.semanticscholar.org/CorpusID:9326192.

[219] Xiaotong Zhang and Jin Zhang. "Subject Independent Human Activity Recognition with Foot IMU Data". In: *2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*. 2019, pp. 240–246.

[220] Young-Seol Lee and Sung-Bae Cho. "Activity recognition using hierarchical hidden markov models on a smartphone with 3D accelerometer". In: *Hybrid Artificial Intelligent Systems: 6th International Conference, HAIS 2011, Wroclaw, Poland, May 23-25, 2011, Proceedings, Part I 6*. Springer. 2011, pp. 460–467.

[221] Tim LM Van Kasteren, Gwenn Englebienne, and Ben JA Kröse. "Hierarchical activity recognition using automatically clustered actions". In: *Ambient Intelligence: Second International Joint Conference on AmI 2011, Amsterdam, The Netherlands, November 16-18, 2011. Proceedings 2*. Springer. 2011, pp. 82–91.

[222] Liang Wang et al. "A hierarchical approach to real-time activity recognition in body sensor networks". In: *Pervasive and Mobile Computing* 8.1 (2012), pp. 115–130. URL: https://www.sciencedirect.com/science/article/pii/S157411921000129X.

[223] Qingbin Gao and Shiliang Sun. "Trajectory-based human activity recognition with hierarchical dirichlet process hidden Markov models". In: *2013 IEEE China Summit and International Conference on Signal and Information Processing*. 2013, pp. 456–460.

[224] Eric Rosen and Doruk Senkal. *CHARM: A Hierarchical Deep Learning Model for Classification of Complex Human Activities Using Motion Sensors*. 2022. URL: https://arxiv.org/abs/2207.07806.

[225] Hui-Huang Hsu et al. "Two-phase activity recognition with smartphone sensors". In: *2015 18th International Conference on Network-Based Information Systems*. IEEE. 2015, pp. 611–615.

[226] Gabriel Filios et al. "Hierarchical algorithm for daily activity recognition via smartphone sensors". In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE. 2015, pp. 381–386.

[227] Charissa Ann Ronao and Sung-Bae Cho. "Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models". In: *International Journal of Distributed Sensor Networks* 13.1 (2017), p. 1550147716683687.

[228] Adil Mehmood Khan et al. "A Triaxial Accelerometer-Based Physical-Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer". In: *IEEE Transactions on Information Technology in Biomedicine* 14.5 (2010), pp. 1166–1172.

[229] Heeryon Cho and Sang Min Yoon. "Divide and Conquer-Based 1D CNN Human Activity Recognition Using Test Data Sharpening". In: *Sensors* 18.4 (2018). URL: https://www.mdpi.com/1424-8220/18/4/1055.

[230] Juan Ye, Graeme Stevenson, and Simon Dobson. "A top-level ontology for smart environments". In: *Pervasive and Mobile Computing* 7.3 (2011). Knowledge-Driven Activity Recognition in Intelligent Environments, pp. 359–378. URL: https://www.sciencedirect.com/science/article/pii/S1574119211000277.

[231] George Okeyo et al. "A Hybrid Ontological and Temporal Approach for Composite Activity Modelling". In: *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. 2012, pp. 1763–1770.

[232] George Okeyo, Liming Chen, and Hui Wang. "Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes". In: *Future Generation Computer Systems* 39 (2014). Special Issue on Ubiquitous Computing and Future Communication Systems, pp. 29–43. URL: https://www.sciencedirect.com/science/article/pii/S0167739X14000399.

[233] Daniele Riboni and Claudio Bettini. "OWL 2 modeling and reasoning with complex human activities". In: *Pervasive and Mobile Computing* 7.3 (2011). Knowledge-Driven Activity Recognition in Intelligent Environments, pp. 379–395. URL: https://www.sciencedirect.com/science/article/pii/S1574119211000265.

[234] Liming Chen, Chris Nugent, and George Okeyo. "An Ontology-Based Hybrid Approach to Activity Modeling for Smart Homes". In: *IEEE Transactions on Human-Machine Systems* 44.1 (2014), pp. 92–105.

[235] Liming Chen, Chris Nugent, and Ahmad Al-Bashrawi. "Semantic data management for situation-aware assistance in ambient assisted living". In: *Proceedings of the 11th international conference on information integration and web-based applications & services*. 2009, pp. 298–305.

[236] Liming Chen, Chris Nugent, and Hui Wang. "A Knowledge-Driven Approach to Activity Recognition in Smart Homes". In: *Knowledge and Data Engineering, IEEE Transactions on* 24 (June 2012), pp. 1–1.

[237] Daniele Riboni and Claudio Bettini. "Context-Aware Activity Recognition through a Combination of Ontological and Statistical Reasoning". In: *Ubiquitous Intelligence and Computing*. Ed. by Daqing Zhang et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 39–53.

[238] Liming Chen, Chris Nugent, and Joseph Rafferty. "Ontology-based Activity Recognition Framework and Services". In: *Proceedings of International Conference on Information Integration and Web-Based Applications & Services*. IIWAS '13. Vienna, Austria: Association for Computing Machinery, 2013, pp. 463–469. URL: https://doi.org/10.1145/2539150.2539187.

[239] Gabriele Civitarese et al. "POLARIS: Probabilistic and Ontological Activity Recognition in Smart-Homes". In: *IEEE Transactions on Knowledge and Data Engineering* 33.1 (2021), pp. 209–223.

[240] Jingkang Yang et al. *Generalized Out-of-Distribution Detection: A Survey*. 2022. arXiv: 2110.11334 [cs.CV].

[241] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1050–1059. URL: https://proceedings.mlr.press/v48/gal16.html.

[242] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf.

[243] Andrey Malinin and Mark Gales. "Predictive Uncertainty Estimation via Prior Networks". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 7047–7058.

[244] Jay Nandy, Wynne Hsu, and Mong Li Lee. "Towards Maximizing the Representation Gap between In-Domain & Out-of-Distribution Examples". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 9239–9250. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/68d3743587f71fbaa5062152985aff40-Paper.pdf.

[245] Shiyu Liang, Yixuan Li, and R. Srikant. *Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks*. 2020. arXiv: 1706.02690 [cs.LG].

[246] Yen-Chang Hsu et al. *Generalized ODIN: Detecting Out-of-distribution Image without Learning from Out-of-distribution Data*. 2020. arXiv: 2002.11297 [cs.CV].

[247] Lukas Ruff et al. "Deep One-Class Classification". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 4393–4402. URL: https://proceedings.mlr.press/v80/ruff18a.html.

[248] Yiyou Sun, Chuan Guo, and Yixuan Li. *ReAct: Out-of-distribution Detection With Rectified Activations*. 2021. arXiv: 2111.12797 [cs.LG].

[249] X. Dong et al. "Neural Mean Discrepancy for Efficient Out-of-Distribution Detection". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2022, pp. 19195–19205. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01862.

[250] Abhijit Bendale and Terrance Boult. *Towards Open Set Deep Networks*. 2015. arXiv: 1511.06233 [cs.CV].

[251] Xin Sun et al. *M2IOSR: Maximal Mutual Information Open Set Recognition*. 2021. arXiv: 2108.02373 [cs.CV].

[252] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. *Exploring the Limits of Out-of-Distribution Detection*. 2021. arXiv: 2106.03004 [cs.LG].

[253] Rui Huang, Andrew Geng, and Yixuan Li. *On the Importance of Gradients for Detecting Distributional Shifts in the Wild*. 2021. arXiv: 2110.00218 [cs.LG].

[254] Rui Huang and Yixuan Li. *MOS: Towards Scaling Out-of-distribution Detection for Large Semantic Space*. 2021. arXiv: 2105.01879 [cs.CV].

[255] Kibok Lee et al. *Hierarchical Novelty Detection for Visual Object Recognition*. 2018. arXiv: 1804.00722 [cs.CV].

[256] Weitang Liu et al. *Energy-based Out-of-distribution Detection*. 2021. arXiv: 2010.03759 [cs.LG].

[257] Haoran Wang et al. *Can multi-label classification networks know what they don't know?* 2021. arXiv: 2109.14162 [cs.LG].

[258] Akshay Raj Dhamija, Manuel Günther, and Terrance E. Boult. *Reducing Network Agnostophobia*. 2018. arXiv: 1811.04110 [cs.CV].

[259] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. *Deep Anomaly Detection with Outlier Exposure*. 2019. arXiv: 1812.04606 [cs.LG].

[260] Aristotelis-Angelos Papadopoulos et al. "Outlier exposure with confidence control for out-of-distribution detection". In: *Neurocomputing* 441 (June 2021), pp. 138–150. URL: http://dx.doi.org/10.1016/j.neucom.2021.02.007.

[261] Christophe Leys et al. "Detecting multivariate outliers: Use a robust variant of the Mahalanobis distance". In: *Journal of Experimental Social Psychology* 74 (2018), pp. 150–156. URL: https://www.sciencedirect.com/science/article/pii/S0022103117302123.

[262] Bo Zong et al. "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection". In: *International Conference on Learning Representations*. 2018. URL: https://api.semanticscholar.org/CorpusID:51805340.

[263] Davide Abati et al. *Latent Space Autoregression for Novelty Detection*. 2019. arXiv: 1807.01653 [cs.CV].

[264] Jingkang Yang, Kaiyang Zhou, and Ziwei Liu. *Full-Spectrum Out-of-Distribution Detection*. 2022. arXiv: 2204.05306 [cs.CV].

[265] Kimin Lee et al. *A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks*. 2018. arXiv: 1807.03888 [stat.ML].

[266] Jie Ren et al. *A Simple Fix to Mahalanobis Distance for Improving Near-OOD Detection*. 2021. arXiv: 2106.09022 [cs.LG].

[267] Joost van Amersfoort et al. *Uncertainty Estimation Using a Single Deep Deterministic Neural Network*. 2020. arXiv: 2003.02037 [cs.LG].

[268] Jingkang Yang et al. *Semantically Coherent Out-of-Distribution Detection*. 2021. arXiv: 2108.11941 [cs.CV].

[269] Yiyou Sun et al. *Out-of-Distribution Detection with Deep Nearest Neighbors*. 2022. arXiv: 2204.06507 [cs.LG].

[270] Mohammad Sabokrou et al. *Adversarially Learned One-Class Classifier for Novelty Detection*. 2018. arXiv: 1802.09088 [cs.CV].

[271] Poojan Oza and Vishal M Patel. *C2AE: Class Conditioned Auto-Encoder for Open-set Recognition*. 2019. arXiv: 1904.01198 [cs.CV].

[272] Xin Sun et al. *Conditional Gaussian Distribution Learning for Open Set Recognition*. 2021. arXiv: 2003.08823 [cs.LG].

[273] Shuyu Lin et al. "Anomaly Detection for Time Series Using VAE-LSTM Hybrid Model". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 4322–4326.

[274] Houssam Zenati et al. *Efficient GAN-Based Anomaly Detection*. 2019. arXiv: 1802.06222 [cs.LG].

[275] Xu Han, Xiaohui Chen, and Li-Ping Liu. *GAN Ensemble for Anomaly Detection*. 2020. arXiv: 2012.07988 [cs.LG].

[276] Fuzhen Zhuang et al. "A Comprehensive Survey on Transfer Learning". In: *Proceedings of the IEEE* 109.1 (2021), pp. 43–76.

[277] Shuteng Niu et al. "A Decade Survey of Transfer Learning (2010–2020)". In: *IEEE Transactions on Artificial Intelligence* 1.2 (2020), pp. 151–166.

[278] Wenyuan Dai et al. "Boosting for Transfer Learning". In: *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. Corvalis, Oregon, USA: Association for Computing Machinery, 2007, pp. 193–200. URL: https://doi.org/10.1145/1273496.1273521.

[279] Robert E. Schapire. "A Brief Introduction to Boosting". In: *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'99. Stockholm, Sweden: Morgan Kaufmann Publishers Inc., 1999, pp. 1401–1406.

[280] Yi Yao and Gianfranco Doretto. "Boosting for transfer learning with multiple sources". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2010), pp. 1855–1862. URL: https://api.semanticscholar.org/CorpusID:15042288.

[281] Joshua Lee, Prasanna Sattigeri, and Gregory Wornell. "Learning New Tricks From Old Dogs: Multi-Source Transfer Learning From Pre-Trained Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/6048ff4e8cb07aa60b6777b6f7384d52-Paper.pdf.

[282] Boyu Wang et al. “Transfer Learning via Minimizing the Performance Gap between Domains”. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[283] Jason Yosinski et al. “How Transferable Are Features in Deep Neural Networks?” In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’14. Montreal, Canada: MIT Press, 2014, pp. 3320–3328.

[284] Muhammad Ghifary, W. Bastiaan Kleijn, and Mengjie Zhang. *Domain Adaptive Neural Networks for Object Recognition*. 2014. arXiv: 1409.6041 [cs.CV].

[285] Hemanth Venkateswara et al. *Deep Hashing Network for Unsupervised Domain Adaptation*. 2017. arXiv: 1706.07522 [cs.CV].

[286] Baochen Sun, Jiashi Feng, and Kate Saenko. “Return of Frustratingly Easy Domain Adaptation”. In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI’16. Phoenix, Arizona: AAAI Press, 2016, pp. 2058–2065.

[287] Baochen Sun and Kate Saenko. *Deep CORAL: Correlation Alignment for Deep Domain Adaptation*. 2016. arXiv: 1607.01719 [cs.CV].

[288] Mingsheng Long et al. *Deep Transfer Learning with Joint Adaptation Networks*. 2017. arXiv: 1605.06636 [cs.LG].

[289] Yaroslav Ganin et al. *Domain-Adversarial Training of Neural Networks*. 2016. arXiv: 1505.07818 [stat.ML].

[290] Wenyuan Dai et al. “Co-Clustering Based Classification for out-of-Domain Documents”. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’07. San Jose, California, USA: Association for Computing Machinery, 2007, pp. 210–219. URL: https://doi.org/10.1145/1281192.1281218.

[291] Wenyuan Dai et al. “Self-taught clustering”. In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 200–207.

[292] L. Duan et al. *Domain adaptation from multiple sources via auxiliary classifiers*. 2009.

[293] Lixin Duan, Dong Xu, and Ivor Wai-Hung Tsang. “Domain adaptation from multiple sources: A domain-dependent regularization approach”. In: *IEEE Transactions on neural networks and learning systems* 23.3 (2012), pp. 504–518.

[294] Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. “Safety in Numbers: Learning Categories from Few Examples with Multi Model Knowledge Transfer”. In: Dec. 2010, pp. 3081–3088.

[295] Zhongtang Zhao et al. “Cross-People Mobile-Phone Based Activity Recognition.” In: Jan. 2011, pp. 2545–2550.

[296] Mingsheng Long et al. *Learning Multiple Tasks with Multilinear Relationship Networks*. 2017. arXiv: 1506.02117 [cs.LG].

[297] Jing Gao et al. “Knowledge Transfer via Multiple Model Local Structure Mapping”. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’08. Las Vegas, Nevada, USA: Association for Computing Machinery, 2008, pp. 283–291. URL: https://doi.org/10.1145/1401890.1401928.

[298] Raymond J. Mooney. “Transfer Learning by Mapping and Revising Relational Knowledge”. In: *Advances in Artificial Intelligence - SBIA 2008*. Ed. by Gerson Zaverucha and Augusto Loureiro da Costa. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 2–3.

[299] Jesse Davis and Pedro Domingos. “Deep Transfer via Second-Order Markov Logic”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML ’09. Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 217–224. URL: https://doi.org/10.1145/1553374.1553402.

[300] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[301] Sainbayar Sukhbaatar et al. *Training Convolutional Networks with Noisy Labels*. 2015. arXiv: 1406.2080 [cs.CV].

[302] Jacob Goldberger and Ehud Ben-Reuven. “Training deep neural-networks using a noise adaptation layer”. In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=H12GRgcxg.

[303] Tong Xiao et al. “Learning From Massive Noisy Labeled Data for Image Classification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[304] Bo Han et al. *Masking: A New Perspective of Noisy Supervision*. 2018. arXiv: 1805.08193 [cs.LG].

[305] Kimin Lee et al. *Robust Inference via Generative Classifiers for Handling Noisy Labels*. 2019. arXiv: 1901.11300 [stat.ML].

[306] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML’15. Lille, France: JMLR.org, 2015, pp. 448–456.

[307] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. *Using Pre-Training Can Improve Model Robustness and Uncertainty*. 2019. arXiv: 1901.09960 [cs.LG].

[308] Xiaobo Xia et al. “Robust early-learning: Hindering the memorization of noisy labels”. In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=Eql5b1_hTE4.

[309] Michal Lukasik et al. “Does Label Smoothing Mitigate Label Noise?” In: *Proceedings of the 37th International Conference on Machine Learning*. ICML’20. JMLR.org, 2020.

[310] Simon Jenni and Paolo Favaro. "Deep Bilevel Learning". In: *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*. Munich, Germany: Springer-Verlag, 2018, pp. 632–648. URL: https://doi.org/10.1007/978-3-030-01249-6_38.

[311] Aditya Krishna Menon et al. "Can gradient clipping mitigate label noise?" In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=rklB76EKPr.

[312] Ryutaro Tanno et al. *Learning From Noisy Labels By Regularized Estimation Of Annotator Confusion*. 2019. arXiv: 1902.03680 [cs.LG].

[313] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: *CoRR* abs/1412.6572 (2014). URL: https://api.semanticscholar.org/CorpusID:6706414.

[314] Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. *Robust Loss Functions under Label Noise for Deep Neural Networks*. 2017. arXiv: 1712.09482 [stat.ML].

[315] Zhilu Zhang and Mert R. Sabuncu. "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 8792–8802.

[316] Amirmasoud Ghiassi, Robert Birke, and Lydia Y. Chen. *Robust Learning via Golden Symmetric Loss of (un)Trusted Labels*. 2021. URL: https://openreview.net/forum?id=20qC5K2ICZL.

[317] Giorgio Patrini et al. *Making Deep Neural Networks Robust to Label Noise: a Loss Correction Approach*. 2017. arXiv: 1609.03683 [stat.ML].

[318] Dan Hendrycks et al. "Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/ad554d8c3b06d6b97ee76a2448bd7913-Paper.pdf.

[319] Xiaobo Xia et al. *Are Anchor Points Really Indispensable in Label-Noise Learning?* 2019. arXiv: 1906.00189 [cs.LG].

[320] Yu Yao et al. *Dual T: Reducing Estimation Error for Transition Matrix in Label-noise Learning*. 2021. arXiv: 2006.07805 [cs.LG].

[321] Scott Reed et al. *Training Deep Neural Networks on Noisy Labels with Bootstrapping*. 2015. arXiv: 1412.6596 [cs.CV].

[322] Eric Arazo et al. "Unsupervised Label Noise Modeling and Loss Correction". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 312–321. URL: https://proceedings.mlr.press/v97/arazo19a.html.

[323] Lang Huang, Chao Zhang, and Hongyang Zhang. *Self-Adaptive Training: beyond Empirical Risk Minimization*. 2020. arXiv: 2002.10319 [cs.LG].

[324] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. "SELFIE: Refurbishing Unclean Samples for Robust Deep Learning". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 5907–5915. URL: https://proceedings.mlr.press/v97/song19b.html.

[325] Eran Malach and Shai Shalev-Shwartz. "Decoupling "When to Update" from "How to Update"". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 961–971.

[326] Lu Jiang et al. "MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 2304–2313. URL: https://proceedings.mlr.press/v80/jiang18c.html.

[327] Bo Han et al. *Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels*. 2018. arXiv: 1804.06872 [cs.LG].

[328] Xingrui Yu et al. "How does Disagreement Help Generalization against Label Corruption?" In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 7164–7173. URL: https://proceedings.mlr.press/v97/yu19b.html.

[329] Hongxin Wei et al. *Combating noisy labels by agreement: A joint training method with co-regularization*. 2020. arXiv: 2003.02752 [cs.CV].

[330] Yanyao Shen and Sujay Sanghavi. "Learning with Bad Training Data via Iterative Trimmed Loss Minimization". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 5739–5748. URL: https://proceedings.mlr.press/v97/shen19e.html.

[331] Pengfei Chen et al. *Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels*. 2019. arXiv: 1905.05040 [cs.LG].

[332] Jinchi Huang et al. "O2U-Net: A Simple Noisy Label Detection Approach for Deep Neural Networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[333] Duc Tam Nguyen et al. *SELF: Learning to Filter Noisy Labels with Self-Ensembling*. 2019. arXiv: 1910.01842 [cs.CV].

[334] Junnan Li, Richard Socher, and Steven C.H. Hoi. "DivideMix: Learning with Noisy Labels as Semi-supervised Learning". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=HJgExaVtwr.

[335] Ramsey Faragher and Robert Harle. "Location Fingerprinting With Bluetooth Low Energy Beacons". In: *IEEE Journal on Selected Areas in Communications* 33.11 (2015), pp. 2418–2428.

[336] Michael J. Christoe et al. "Bluetooth Signal Attenuation Analysis in Human Body Tissue Analogues". In: *IEEE Access* 9 (2021), pp. 85144–85150.

[337] Xinyu Hou and Jeroen Bergmann. "Pedestrian Dead Reckoning With Wearable Sensors: A Systematic Review". In: *IEEE Sensors Journal* 21.1 (2021), pp. 143–152.

[338] *Seven Different Types of Privacy*. https://www.isss.org.uk/2024/02/02/seven-different-types-of-privacy/. Accessed: 2025-02-09.

[339] *Understanding Different Types of Data about People*. https://www.lboro.ac.uk/data-privacy/help/personaldata/. Accessed: 2025-02-09.

[340] Sungil Kim et al. "Indoor positioning system techniques and security". In: *2015 Forth International Conference on e-Technologies and Networks for Development (ICeND)*. 2015, pp. 1–4.

[341] Raúl Casanova-Marqués et al. "Maximizing privacy and security of collaborative indoor positioning using zero-knowledge proofs". In: *Internet of Things* 22 (2023), p. 100801. URL: https://www.sciencedirect.com/science/article/pii/S2542660523001245.

[342] Yerkezhan Sartayeva and Henry C. B. Chan. "A survey on indoor positioning security and privacy". In: *Computers & Security* 131 (2023), p. 103293. URL: https://www.sciencedirect.com/science/article/pii/S0167404823002031.

[343] A.R. Jiménez et al. "Location of Persons Using Binary Sensors and BLE Beacons for Ambient Assitive Living". In: *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2018, pp. 206–212.

[344] Ming-DE Chen et al. "Accuracy of Wristband Activity Monitors during Ambulation and Activities". en. In: *Med Sci Sports Exerc* 48.10 (Oct. 2016), pp. 1942–1949.

[345] Frederik Rose Svarre et al. "The validity of activity trackers is affected by walking speed: the criterion validity of Garmin Vivosmart(®) HR and StepWatch(™) 3 for measuring steps at various walking speeds under controlled conditions". en. In: *PeerJ* 8 (July 2020), e9381.

[346] Sohrab Saeb, Konrad Körding, and David C. Mohr. "Making Activity Recognition Robust against Deceptive Behavior". In: *PLOS ONE* 10.12 (Dec. 2015), pp. 1–12. URL: https://doi.org/10.1371/journal.pone.0144795.

[347] Nishkam Ravi et al. "Activity recognition from accelerometer data". In: *Aaai*. Vol. 5. 2005. Pittsburgh, PA. 2005, pp. 1541–1546.

[348] Mustafa A. Ghazi et al. "Accelerometer Thresholds for Estimating Physical Activity Intensity Levels in Infants: A Preliminary Study". In: *Sensors* 24.14 (2024). URL: https://www.mdpi.com/1424-8220/24/14/4436.

[349] Treuth MS et al. "Defining accelerometer thresholds for activity intensities in adolescent girls". In: *Medicine and science in sports and exercise* 36.7 (2004), pp. 1259–1266.

[350] Bezuidenhout LJ Moulaee Conradsson D. "Establishing Accelerometer Cut-Points to Classify Walking Speed in People Post Stroke". In: *Sensors (Basel)* 22.11 (2022). URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9185353/.

[351] Buke Ao et al. “Context Impacts in Accelerometer-Based Walk Detection and Step Counting”. In: *Sensors* 18.11 (2018). URL: https://www.mdpi.com/1424-8220/18/11/3604.

[352] Marcin Straczkiewicz, Emily J. Huang, and Jukka-Pekka Onnela. “A “one-size-fits-most”walking recognition method for smartphones, smartwatches, and wearable accelerometers”. In: *npj Digital Medicine* 6.1 (2023), p. 29.

[353] Matthew William Flood, Ben P. F. O’Callaghan, and Madeleine M. Lowery. “Gait Event Detection From Accelerometry Using the Teager–Kaiser Energy Operator”. In: *IEEE Transactions on Biomedical Engineering* 67 (2020), pp. 658–666. URL: https://api.semanticscholar.org/CorpusID:172136687.

[354] Moustafa Alzantot and Moustafa Youssef. “UPTIME: Ubiquitous pedestrian tracking using mobile phones”. In: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. 2012, pp. 3204–3209.

[355] Ducharme SW et al. “A Transparent Method for Step Detection using an Acceleration Threshold”. In: *Journal for the measurement of physical behaviour* 4.4 (2021), pp. 311–320.

[356] Elin Johansson et al. “Sitting, standing and moving during work and leisure among male and female office workers of different age: a compositional data analysis”. In: *BMC Public Health* 20.1 (2020), p. 826.

[357] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[358] Yu Zhao et al. “Deep Residual Bidir-LSTM for Human Activity Recognition Using Wearable Sensors”. In: *CoRR* abs/1708.08989 (2017). arXiv: 1708.08989. URL: http://arxiv.org/abs/1708.08989.

[359] Wu CY et al. “In-Home Mobility Frequency and Stability in Older Adults Living Alone With or Without MCI: Introduction of New Metrics”. In: *Frontiers in digital health* 3 (2021).

[360] *What is the Average Amount of Time We REALLY Spend at Home?* https://www.simplyproductive.com/2022/10/what-is-the-average-amount-of-time-we-really-spend-at-home/. Accessed: 2025-02-09.

[361] Owen N. et al. “Sedentary behavior: emerging evidence for a new health risk.” In: *Mayo Clinic proceedings* 85.12 (2010), pp. 1138–1141.

[362] Bassett D. R. Jr et al. “Pedometer-measured physical activity and health behaviors in U.S. adults”. In: *Medicine and science in sports and exercise* 42.10 (2010), pp. 1819–1825.

[363] *10,000 steps a day: Too low? Too high?* https://www.mayoclinic.org/healthy-lifestyle/fitness/in-depth/10000-steps/art-20317391. Accessed: 2025-02-09.

[364] Lund Rasmussen C. et al. “Light-intensity physical activity derived from count or activity types is differently associated with adiposity markers”. In: *Scandinavian journal of medicine & science in sports* 30.10 (2020), pp. 1966–1975.

[365] *How fast is fast enough? About 100 steps per minute might be a reasonable goal, but your mileage may vary.* https://www.health.harvard.edu/heart-health/walk-this-way. Accessed: 2025-02-09.

[366] Dallan Byrne et al. "Residential wearable RSSI and accelerometer measurements with detailed location annotations". In: *Scientific Data* 5.1 (2018), p. 180168.

[367] Muhammad Shoaib et al. "Fusion of Smartphone Motion Sensors for Physical Activity Recognition". In: *Sensors* 14.6 (2014), pp. 10146–10176. URL: https://www.mdpi.com/1424-8220/14/6/10146.

[368] Pierluigi Casale, Oriol Pujol, and Petia Radeva. "Human Activity Recognition from Accelerometer Data Using a Wearable Device". In: *Proceedings of the 5th Iberian Conference on Pattern Recognition and Image Analysis*. IbPRIA'11. Las Palmas de Gran Canaria, Spain: Springer-Verlag, 2011, pp. 289–296.

[369] Biying Fu et al. "Sensing Technology for Human Activity Recognition: A Comprehensive Survey". In: *IEEE Access* 8 (2020), pp. 83791–83820.

[370] Kuan Zhang et al. "HuAc: Human Activity Recognition Using Crowdsourced WiFi Signals and Skeleton Data". In: *Wireless Communications and Mobile Computing* 2018 (2018), p. 6163475.

[371] Wei Wang et al. "Device-Free Human Activity Recognition Using Commercial WiFi Devices". In: *IEEE Journal on Selected Areas in Communications* 35.5 (2017), pp. 1118–1131.

[372] Zheqi Yu et al. "Data Fusion for Human Activity Recognition Based on RF Sensing and IMU Sensor". In: *Body Area Networks. Smart IoT and Big Data for Intelligent Health Management*. Ed. by Masood Ur Rehman and Ahmed Zoha. Cham: Springer International Publishing, 2022, pp. 3–14.

[373] Tim Weißker et al. "ShakeCast: using handshake detection for automated, setup-free exchange of contact data". In: Sept. 2017, pp. 1–8.

[374] Jie Wang et al. "Device-free simultaneous wireless localization and activity recognition with wavelet feature". In: *IEEE Transactions on Vehicular Technology* 66.2 (2016), pp. 1659–1669.

[375] Harish Haresamudram, David V. Anderson, and Thomas Plötz. "On the role of features in human activity recognition". In: *Proceedings of the 2019 ACM International Symposium on Wearable Computers*. ISWC '19. London, United Kingdom: Association for Computing Machinery, 2019, pp. 78–88. URL: https://doi.org/10.1145/3341163.3347727.

[376] Diane J Cook and Narayanan C Krishnan. *Activity learning: discovering, recognizing, and predicting human behavior from sensor data*. John Wiley & Sons, 2015.

[377] Hendrik Mende et al. "On the importance of domain expertise in feature engineering for predictive product quality in production". In: *Procedia CIRP* 118 (2023). 16th CIRP Conference on Intelligent Computation in Manufacturing Engineering, pp. 1096–1101. URL: https://www.sciencedirect.com/science/article/pii/S2212827123004158.

[378] Peter H Veltink et al. “Detection of static and dynamic activities using uniaxial accelerometers”. In: *IEEE Transactions on Rehabilitation Engineering* 4.4 (1996), pp. 375–385.

[379] Allahbakhshi H. et al. “Physical Activity Type Detection Using Real-Life Data: A Systematic Review”. In: *Frontiers in physiology* 10.75 (2019).

[380] R Adaskevicius. “Method for recognition of the physical activity of human being using a wearable accelerometer”. In: *Elektronika ir Elektrotechnika* 20.5 (2014), pp. 127–131.

[381] Abhijit Suprem, Vishal Deep, and Tarek Elarabi. “Orientation and displacement detection for smartphone device based imus”. In: *IEEE Access* 5 (2016), pp. 987–997.

[382] Nestor Michael Tiglao et al. “Smartphone-based indoor localization techniques: State-of-the-art and classification”. In: *Measurement* 179 (2021), p. 109349.

[383] Man-Yee Kan et al. “How do older adults spend their time? Gender gaps and educational gradients in time use in East Asian and western countries”. In: *Journal of Population Ageing* 14.4 (2021), pp. 537–562.

[384] Gábor Csizmadia et al. “Human activity recognition of children with wearable devices using LightGBM machine learning”. In: *Scientific Reports* 12.1 (2022), pp. 1–10.

[385] Peter Jaksa. *The Importance of a Daily Schedule for Kids with ADHD: Sample Routines and More*. Nature Publishing Group, 2022. URL: `https://tinyurl.com/272xwrzk`.

[386] Norman Wolkove et al. “Sleep and aging: 1. Sleep disorders commonly found in older people.” eng. In: *CMAJ* 176.9 (Apr. 2007), pp. 1299–1304.

[387] Barbara Mostacci et al. “Incidence of sudden unexpected death in nocturnal frontal lobe epilepsy: a cohort study.” eng. In: *Sleep Med* 16.2 (Feb. 2015), pp. 232–236.

[388] Suneth Ranasinghe, Fadi Al Machot, and Heinrich C Mayr. “A review on applications of activity recognition systems with regard to performance and evaluation”. In: *International Journal of Distributed Sensor Networks* 12.8 (2016), p. 1550147716665520.

[389] Arnaud Ahouandjinou, Cina Motamed, and Eugène Ezin. “Temporal and Hierarchical HMM for Activity Recognition Applied in Visual Medical Monitoring using a Multi-Camera System”. In: *revue africaine de la recherche en informatique et mathématiques appliquées* vol 21, (Dec. 2015), pp. 49–66.

[390] Svebor Karaman et al. “Hierarchical Hidden Markov Model in Detecting Activities of Daily Living in Wearable Videos for Studies of Dementia”. In: *Multimedia Tools and Applications* 69.3 (Apr. 2014), pp. 743–771. URL: `https://hal.archives-ouvertes.fr/hal-00639014`.

[391] Jennifer L. Hicks et al. “Best practices for analyzing large-scale health data from wearables and smartphone apps”. In: *npj Digital Medicine* 2.1 (2019), p. 45.

[392] Hwanjun Song et al. *Learning from Noisy Labels with Deep Neural Networks: A Survey*. 2022. arXiv: `2007.08199 [cs.LG]`.

[393] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. "Recognizing Detailed Human Context in the Wild from Smartphones and Smartwatches". In: *IEEE Pervasive Computing* 16.4 (2017), pp. 62–74.

[394] Juan Ye. "SLearn: Shared learning human activity labels across multiple datasets". In: *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 2018, pp. 1–10.

[395] Juan Ye, Simon Dobson, and Franco Zambonelli. "XLearn: Learning Activity Labels across Heterogeneous Datasets". In: *ACM Trans. Intell. Syst. Technol.* 11.2 (Jan. 2020). URL: https://doi.org/10.1145/3368272.

[396] Xuefeng Liang, Xingyu Liu, and Longshan Yao. "Review–A Survey of Learning from Noisy Labels". In: *ECS Sensors Plus* 1.2 (2022).

[397] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.