School of Doctorate University of Milano-Bicocca
Department of Informatics, Systemistics and Communications
Ph.D Program in Computer Science - XXXIII Cycle

Personalization of Human Activity Recognition Methods using Inertial Data

Anna Ferrari
Mtr. 700978

Supervisor: Daniela Micucci
Co-Supervisor: Paolo Napoletano
Tutor: Vincenzina Messina
Ph.D Coordinator: Leonardo Mariani

Academic Year 2020 - 2021

Abstract

Recognizing human activities and monitoring population behavior are fundamental needs of our society. Population security, crowd surveillance, healthcare support and living assistance, lifestyle and behavior tracking are some of the main applications which require the recognition of activities. Activity recognition involves many phases, i.e. the collection, the elaboration and the analysis of information about human activities and behavior. These tasks can be fulfilled manually or automatically, even though a human-based recognition system is not long-term sustainable and scalable.

Nevertheless, transforming a human-based recognition system to computer-based automatic system is not a simple task because it requires dedicated hardware and a sophisticated engineering computational and statistical techniques for data preprocessing and analysis. Recently, considerable changes in technologies are largely facilitating this transformation. Indeed, new hardwares and softwares have drastically modified the activity recognition systems. For example, Micro-Electro-Mechanical Systems (MEMS) progress has enabled a reduction in the size of the hardware. Consequently, costs have decreased. Size and cost reduction allows to embed sophisticated sensors into simple devices, such as phones, watches, and even into shoes and clothes, also called wearable devices. Furthermore, low costs, lightness, and small size have made wearable devices' highly pervasive and accelerated their spread among the population. Today, a very small part of the world population doesn't own a smartphone. According to Digital 2020: Global Digital Overview[1], more than 5.19 billion people now use mobile phones. Among the western countries, smartphones and smartwatches are gadgets of people everyday life.

The pervasiveness is an undoubted advantage in terms of data generation. Huge amount of data, that is big data, are produced every day. Furthermore, wearable devices together with new advanced software technologies enable data to be sent to servers and instantly analyzed by high performing computers.

[1] https://datareportal.com/reports/digital-2020-global-digital-overview

The availability of big data and new technology improvements, permitted Artificial Intelligence models to rise. In particular, machine learning and deep learning algorithms are predominant in activity recognition.

Together with technological and algorithm innovations, the Human Activity recognition (HAR) research field has born. HAR is a field of research which aims at automatically recognizing people's physical activities. HAR investigates on the selection of the best hardware, e. g. the best devices to be used for a given application, on the choice of the software to be dedicated to a specific task, and on the increasing of the algorithm performances.

HAR has been a very active field of research for years and it is still considered one of the most promising research topic for a large spectrum of applications. In particular, it remains a very challenging research field for many reasons. The selection of devices and sensors, the algorithm's performances, the collection and the preprocessing of the data, all are requiring further investigation to improve the overall activity recognition system performances.

In this work, two main aspects have been investigated:

- the benefits of *personalization* on the algorithm performances, when trained on small size datasets: one of the main issue concerning HAR research community is the lack of the availability of public dataset and labelled data. That is, even though the technologies, such as smartphones and wearable devices potentially facilitate the collection of data, the lack of large labelled datasets still remains a predominant issue. Since the algorithms performance hardly depends on the dataset size, many studies have faced this issues by exploiting different personalization definitions. In general, including subject's metadata in the classification, such as age, gender, weight, height, lifestyle, and physical abilities improves the algorithm capability to classify a new instance, even when it is trained on small datasets.

- a comparison of the performances in HAR obtained both from traditional and personalized machine learning and deep learning techniques. In the recent years, machine learning and deep learning techniques have spread in many different field, among them HAR, showing very promising results. We defined and evaluated two novel models: Personalized Machine Learning (PML) and Personalized Deep Learning (PDL) models. We compared them to traditional Machine Learning (ML) and Deep Learning (DL) models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Recognizing human activities and monitoring population behavior are fundamental needs of our society. It is astonishing how many sectors of our socio-economical system are more and more investing in resources, employees, instruments, methods, and techniques, for these purposes.

For instance, security and surveillance detect people's behavior and anomalous activities as well as tracking suspects, victims and witnesses in police investigations. In contexts such as supermarkets, museums, airports, or even cities it is essential to monitor human and crowd behavior to assure high level of security, to prevent thefts and robberies, to organize and manage queues, and to provided smart solutions to optimize journey time. In tactical scenarios, precise information on the soldiers' activities along with their locations and health conditions, is highly beneficial for their performance and safety. Such an information is also helpful to support decision making in both combat and training scenarios. In the hospital environment, invalid patients have to be daily assisted and continuously monitored. In the home environment, activity monitoring helps to complete some actions such as taking medicine, helps in assistance and rehabilitation, and permits detection or even prevention of accidents. Activities monitoring regards also the population lifestyle, human daily behavior, changes in behavior, which can be used for social intervention as well as for commercial purposes. Last, but not least, monitoring people becomes essential in a pandemic scenario, for example to manage with social distance, to monitor and avoid assemblies, to support and stimulate people which may stop to do physical activities during isolation.

Monitoring information about human activities and behavior can be manual or automatic. In general, employing a person to monitor other persons' activity seems to be one simple way, but constant monitoring is not realistic [55]. For instance, in huge and complex contexts, such as airports or supermarkets, or in intimate places, such as homes and hospitals, automatic monitoring becomes essential, or even the only possible solution. These scenarios require huge costs for manual monitoring which are drastically decreased by using automatic engines. A particular example is related to the healthcare systems. The World Health Organization reports that expectancy of life is dramatically increased in the last decades: "a child born in Brazil or Myanmar in 2015 can expect to live 20 years longer than one born in those countries just 50 years ago". Figure 1.1 shows the population growth over the future years.

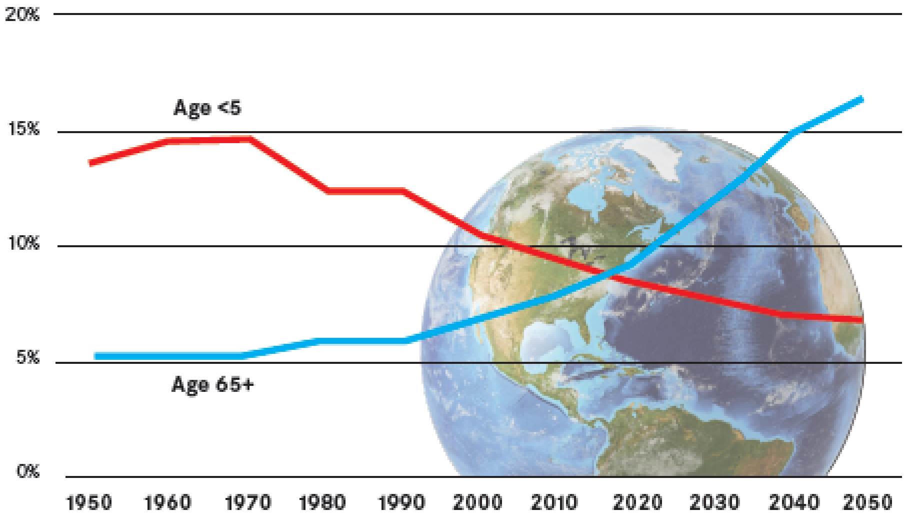


Figure 1.1: Youngh children and older people as a percentage of the global population: 1950 - 2050. Source: World Population Prospects: the 2010 Revision, UN.

Aged population in high income countries presents high rate of neurodegenerative and non-communicable diseases which typically need to be constantly monitored. As a consequence, an aging population increases the number of hospitals' access, resources for assistance and the need of investments for rehabilitation. In this context, a manual recognition system becomes unsustainable since the rate between elderly and worker population is drastically decreasing. The possibility to rely on an automatic and remote monitoring presents twofold benefits; on one hand it substantially reduces the healthcare costs. On the other hand, it improves the patients' life quality and their independence.

Transforming a human-based recognition system to computer-based automatic recognition system is not a simple task. Recognizing an activity or understanding a situation are relative easy for humans but become extremely complex for computers. Indeed, automatic recognition systems require both a dedicated hardware and a sophisticated engineering computational and statistical techniques for data preprocessing and analysis.

Human Activity Recognition (HAR) is the field of research which focuses on all these tasks. HAR automatically recognizes human activity by analyzing signals acquired by sensors. Signals are usually acquired through two main typologies of devices: environmental and wearable. Among the environmental devices, cameras are the most used, while wearable devices encompass all on body worn sensors, such as smart-shirt, smart-shoes, ad-hoc Inertial Measurement Unit (IMU), smartphones and smartwatches.

It is not possible to define a preference between using environmental or wearable devices. The choice depends on the specific application domain. For instance, cameras are preferable in security, surveillance and assistance living scenarios where the monitoring is based on indoor scenarios and on the interactions between users, environment and objects together. However, environmental devices are not spread among the population because they need to be installed and maintained, which often results in high costs. They are also often perceived intrusive in terms of privacy [131]. In contrast, wearables are preferable for outdoor applications, even though they present several limitations in terms of energy consumption, computation capability, among others.

Nevertheless, wearable devices have gained more and more attention from the HAR research community for many reasons. First, they show more flexibility in terms of portability. Indeed, they can be carried outdoor, they do not have to be installed and most of them are part of user's daily life. Wearable enables to receive information about user context, and are normally perceived less intrusive in terms of privacy. Second, the micro-electro-mechanical systems (MEMS) technology evolution has reduced sensor size, cost, and power needs, while sensor's capacity, precision and accuracy have increased. Wearable devices are usually equipped with many micro-sensors, such as accelerometer, gyroscope, GPS, and can be easily integrated with external sensors. Thanks to these sensors, they are able to record many information about user's daily life without being invasive.

Recent hardware and software technologies allow modern wearable devices to perform multitasks activities. Nowadays, a smartphone is able to capture motion, location, and estimate user's activities, while he\she is reading an email. Low cost, high performing technology, and portability lead wearables to drastically spread among the population. Among wearables, smartphones became the most used devices in human's daily life [38].

According to Digital 2020: Global Digital Overview[1], more than 5.19 billion people now use mobile phones. Figure 1.2 show the distribution in how we use the smartphone applications. People use apps in almost every aspect of their lives, whether it's staying in

[1] https://datareportal.com/reports/digital-2020-global-digital-overview

touch with friends and family, relaxing on the couch, managing their finances, getting fit, or even finding love. To be pointed out is that the population uses application for maps 65%, games 47%, and health and fitness 26% which mostly base on HAR technologies 1.2.

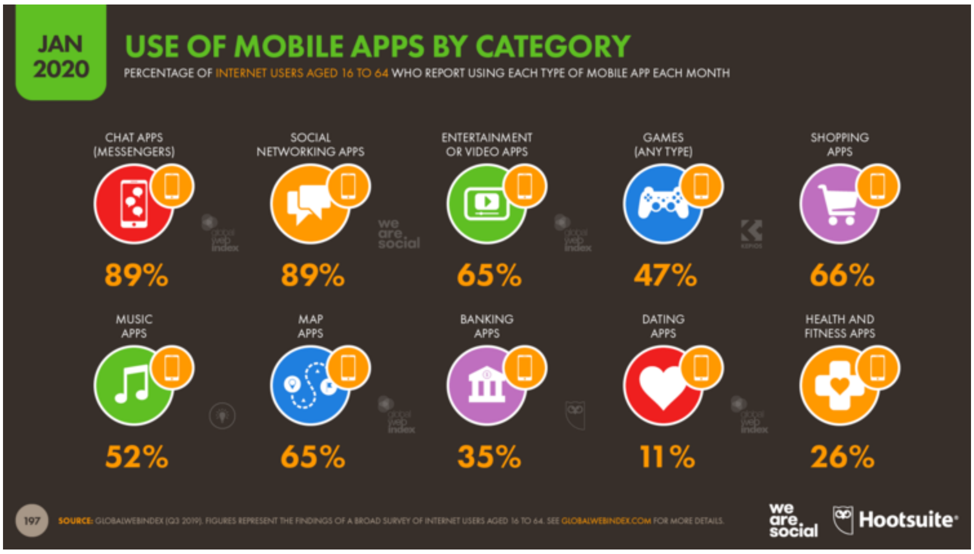


Figure 1.2: Use of Mobile Apps by Category January 2020. Source: DataReportal.

The growing pervasiveness results in the possibility to generate high amount of data, often characterized as *big data*. Availability of big data and technology's improvements, permitted Artificial Intelligence models to rise.

In terms of algorithms, supervised machine learning and deep learning methods are predominant in HAR [55].

Both techniques are valid and powerful and have been largely investigated in the literature. Nevertheless, the application of these techniques remains challenging. In particular, traditional machine learning and deep learning techniques are limited in their ability to generalize to new users and/or new environments, and require considerable effort and customization to achieve good performance in a real-context. One of the most relevant difficulties to face with new situations is due to the population diversity problem, that is, the natural differences between users' activity patterns, which implies that different executions of the same activity are different. In particular two factors influence why the same activity is carried out in a different way [151]:

- *Inter-subject variability*, which refers to anthropometric differences of body parts or to incongruous personal styles in accomplishing the scheduled action. In other words, it refers to the intrinsic differences between subjects in performing the same activity. For instance, acceleration signal of a young woman's walk is different from old man one.

- *Intra-subject variability*, which represents the random nature of a single action class and reflects the fact that the same subject never performs an action in the same way.

Ideally, algorithms should be trained on a representative number of subjects and on as many cases as possible. The number of subjects present in the data set does not just impact the quality and robustness of the induced model, but also the ability to evaluate the consistency of results across subjects [83]. When the availability of the data is scarce and limited the responsibility of generalization is left to the activity classification models which should be able to adapt as much as possible with respect to the final user.

One solution is represented from the personalized models. Personalized models encompass all the techniques which extract additional information directly from the user's metadata or from context's sources to complete and reinforce the algorithm's training. In the state-of-the-art different approaches using users-based information have have been explored. These personalization approaches can be split into three groups: data-based, classifier-based, and similarity-based personalization.

The *data-based approach* bases on different selections of training and test dataset and encompasses: subject-independent, subject-dependent, and hybrid. The subject-independent model does not use the end user data for the training of the activity recognition model. The subject-dependent model only uses the end user data for the development of the activity recognition model. The hybrid model uses the end user data and the data of the other users for the development of the activity recognition model. These three splits, aim at capturing in different ways the influence of including or excluding the test user in the training procedure of the classification.

Classifier-based approach obtains generalization from several and weighed combinations of activity recognition models which permits to achieve better activity recognition performance for the final user.

Similarity-based approach which consider the similarity between users as crucial factor for obtaining a classification models able to adapt to new situations. In particular, studies demonstrated that different physical characteristics are associated to different data patterns, for the same activity. Consequently, further user's information are able to better address the classification decision.

Taking inspiration from the state-of-the art personalization approaches, we developed novel personalized machine learning and personalized deep learning models to improve traditional machine learning and deep learning techniques in terms of generalization capability. This work aims at showing how we implemented these models and at comparing these model with benchmark results.

All models have been trained and tested on public datasets to assure the reproducibility of the results.

1.2 Contributions

The main contributions of this work focus on the improvement of traditional machine learning and deep learning techniques, based on the personalization approaches. In particular, the following points summarize the crucial investigation of the work:

- *personalized machine learning models*: novel classification models based on the personalization of machine learning techniques have been proposed. In particular, machine learning algorithms have been integrated with a weights matrix, called similarity matrix, based on three main user's characteristics: physical, signal, and a combination of both. Each element $sim(i, j)$ of the similarity matrix spans between 0 and 1 and represent the similarity between the subject i and j, reliant to the three above mentioned characteristics. The most similar the subjects are, the closest to 1 is the value of the element $sim(i, j)$, the most the subject i data counts for the classification of the user i. Different machine learning classifier have been tested, and between Support Vector Machine, k-NN and Adaboost, the latter have show more flexibility and best results.

- *comparison between machine learning and deep learning techniques*: since large scale inertial datasets are not available, it is therefore not obvious which method between deep and traditional machine learning methods is the most appropriate. We implement machine learning and deep learning models varying between different input features. Support Vector Machine, k-NN and Residual Neural Network have been compared. The experiments have been based on unimodal and multimodal sensors data. Results demonstrate high robustness in terms of input data, and overall better performance using deep learning techniques.

- *comparison between personalized machine learning and deep learning techniques*: a comparison between deep learning and personalized machine learning methods have been implemented with the aim at investigating the robustness of the deep learning techniques in terms of intra and inter variability across subjects. Results demonstrated that deep learning accuracy outperforms personalized machine learning accuracy in most of the cases.

- *personalized deep learning techniques and comparison with personalized machine learning*: novel classification models based on the personalization of deep learning techniques have been proposed. Along the line of the personalized machine learning models, the similarity matrix has been exploited to select the most similar subject's data, with respect to the test user, as training dataset for an end-to-end deep learning. The comparison with personalized machine learning models doesn't show a clear difference between personalization-based machine learning and deep learning models.

All algorithms have been trained and tested on public dataset for guaranteeing the reproducibility of the results.

1.3 Outline

The rest of the work is organized as follows:

- Chapter 2 gives an overview about the state-of-the-art in Human Activity Recognition. Devices and sensors, methods and techniques, data elaboration and public datasets are discussed in the context of HAR
- Chapter 3 discusses the importance to build personalized machine learning models, and shows how personalized machine learning models have to be implemented and the results in comparison with state-of-the-art approaches. An introductory data preprocessing procedure is described
- Chapter 4 aims at comparing deep learning techniques against machine learning. In particular, we compared deep learning with traditional machine learning and with personalized machine learning. Furthermore, we propose a personalized deep learning method and we compared its performances with machine learning methods.
- Chapter 5 presents a final discussion about the results of the study and sketches the conclusions.

1.4 Acknowledgements

I would like to thank particularly Daniela Micucci, my supervisor, for guiding me in this journey and for giving me precious advices about research as well as about life.

I would like to thank Daniela Micucci and Paolo Napoletano responsible of making me become the researcher I am today, for supporting and encouraging me during these three years.

I would like to thank Carlo Batini for believing in me and proposing me great projects in a more didactical world. It has been an honor working with him.

I thank my office mates of SAL, Marco, Davide and Alessandro for making my working days greater.

I would like to thank all colleagues at CERN - Openlab for the fantastic year spent together, and for their friendship.

A mention is due to my colleagues at IVL, colleagues with which I started my academic journey and with which I hope to collaborate again.

Chapter 2

HAR State-of-the-art

The first work on human activity recognition date back to the late '90s [47]. During the last 30 years, the Human Activity Recognition research community has been very active and has produced methods, techniques, results and datasets in abundance. Additionally, last hardware and software technology improvement together with the continuously increasing pervasiveness of low cost devices, rises the role of HAR in many research and business contexts, such as surveillance, healthcare, delivering, among others.

In the context of HAR, a precise protocol called Activity Recognition Process (ARP) is defined and illustrated in Figure 2.1. The ARP is composed of four phases, acquisition, preproccessing, segmentation, and feature extraction.

In *data acquisition* phase, a receiver obtains data from sensors located in different part of the body or embedded in a device. In smartphones, data from acceleration, angular velocity, magnetic field are normally recorded and stored into files. Files are then transfert to a computer to be elaborated. Data coming from sensors typically include artifacts and noises due to many reasons, such as electronic fluctuation, sensors calibration and malfunctions and have to be processed. The *preprocessing* phases is responsible for the elimination of artifacts and noise. Generally, preprocessing is based on filtering techniques.

In HAR literature, Butterworth low-pass filter is widely used [7, 23, 125]. It is stated that the cut-off frequency of 15Hz is enough to capture human body motion [9]. After having been filtered, data pass to the *data segmentation* phase. Data segmentation is a process responsable to split data into segments, also called windows. Windows can be of different size, normally expressed in seconds. They may contain different number of value, depending on the sample rate. Other typologies of data segmentation are used in HAR, see 2.2.2.1 for further details.

Data segmentation is a common practice which facilitates feature extraction phase.The *features extraction* phase aims at extracting the more important information from the data to be given to the classification algorithm, while reducing data dimensionality. The *classification* is the last phase of the process. It consists in training and testing the algorithm. That is, the parameters of the classification model are estimated during the training procedure. Thereinafter, the classification performances of the model are tested in the testing procedure, see 2.3.2.3.

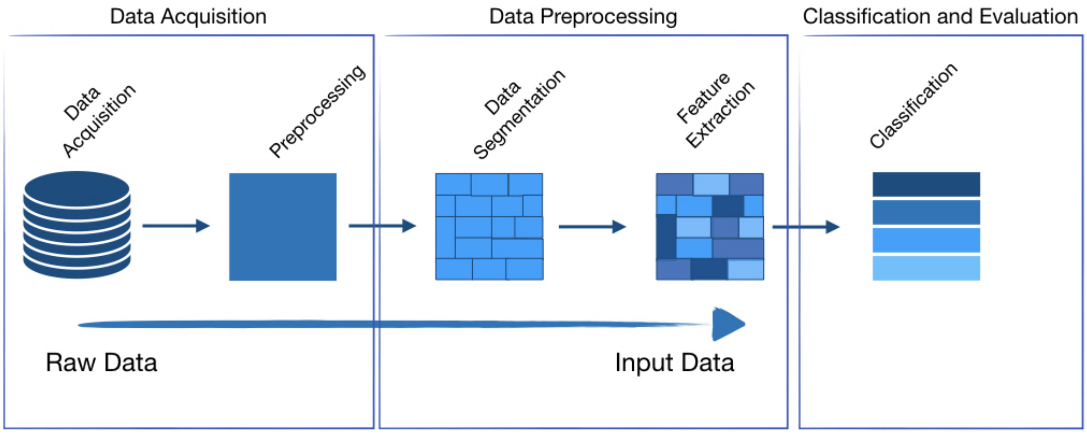


Figure 2.1: Activity Recognition Process: All phases, from data acquisition to classification and evaluation.

In this work, we didn't perform data acquisition and preprocessing phases because usually data provided from public datasets have been already passed these phases.

This Chapter is a survey concerning the last state-of-the-art HAR community trends about all the phases of the ARP. The analysis of the state-of-the-art encompasses scientific articles and papers selected based on the following criteria and keywords:

- first 100 papers found in Google Scholar with key words: **human activity recognition smartphone**,
- first 100 papers found in Google Scholar with key words: **human activity recognition smartphone** starting from 2015,
- first 100 papers found in Google Scholar with key words: **personalized human activity recognition smartphone**,
- first 100 papers found in Google Scholar with key words: **personalized human activity recognition smartphone** staring from 2015

This Chapter is organized as follows. In Section 2.1 we describe devices and sensors exploited in HAR for data acquisition with a particular focus on wearable devices and embedded sensors. We explain why in this work we selected smartphone's data for our analysis, and consequently, why they are preferable among other devices. In Section 2.2, we discuss about benchmark datasets and the data elaboration, i.e. the data segmentation and the feature extraction. In Section 2.3, we describe the most recent classification methods, their strength and weakness. We explain the concept of personalization and why the personalization of those techniques is necessary to improve the overall classification performance.

2.1 Data Acquisition: Devices and Sensors

Over the past decade, a considerable progress in hardware and software technologies has modified habits of the entire population and business. On one hand, the micro-electro-mechanical systems (MEMS) have reduced sensors size, costs, and power needs of sensors, while capacity, precision and accuracy have increased. On the other hand, the spread of the Internet of Things (IoT) has enabled the spread of easy and fast connections between devices, objects and environments.

The pervasiveness and the reliability of these new technologies enable, nowadays, the acquisition and the storage of a large amount of multimodal data [100]. Most recent devices are extremely interconnected, accessible to people and handsome in terms of capability to collect and share large amount of data very quickly. Smartphones, smartwatches, home assistants, drones are daily used and represent essential instruments for many economy business, such as remote healthcare, merchandise delivering, agricoltore, and others. New technologies together with large availability of data gained the attention from research community, including human activity recognition.

The goal of this Section is to present the most used devices for data acquisition in HAR. According to [73], two main categories of devices are generally used in this context: *environmental* and *wearable* devices, see following lines for further details.

2.1.1 Environmental Devices and Sensors

Environmental devices are fixed in predetermined points of interest, so the inference of activities entirely depends either on their location, for instance in case of cameras, or on the voluntary interaction of the users with the sensors, for instance in case of sensors placed on objects.

Environmental devices and sensors are historically related to the definition of environmental monitoring. According to [94, 37], the environmental monitoring was used to measure physical environmental parameters, such as temperature, humidity, pressure. Progressively with the technology development, the monitoring of other environmental parameters became more accessible and easier to acquire, i.e. the number of people inside the environment and their position, the position and the actions performed inside the environment. More recently environmental devices are mostly used to detect the interaction between users and environment. *Cameras*, for instance, generated video sequences or digitalized visual data which are largely used in human activity recognition for surveillance [56], understanding dynamic scene activity [27], and assisted living [96, 116, 147, 54]. *WiFi* is a local-area wireless network connection technology which uses a transmitter to send signals to a receiver. Since human bodies are good reflectors of wireless signals, human activities can be recognized by monitoring changes in WiFi signals [139]. Other environmental technologies mentioned in human activity recognition are *Radio-frequency identification (RFID)* which is based on using electromagnetic fields to automatically identify and track the tags embedded in everyday objects, which contains electronically stored information[25], and *Radar* which uses transmitters and antennas which are mounted on the same side of users. Doppler effect is the basis of the radar-based system[78].

There are several advantages in exploiting environmental devices for human activity recognition as discussed in the following. First, they are much more proficient to recognize complex activities, such as eating, drinking, having a shower, teeth brushing, and others, because data are related to the interaction between many object sensors and the user. Second, environmental devices enable continuous monitoring independently from a battery and from the user. Third, they can be used to detect actions and interaction of multi-residents simultaneously. Finally, environmental devices outperform wearables in terms of in-door localization efficacy.

Nonetheless, nothing can be done if the user is out of the reach of the sensors or they perform activities that do not require interaction with them. However, all outdoor activities cannot be monitored or recognized by environmental devices. In addition, privacy intrusiveness and pervasiveness make difficult to let environmental devices to be totally accepted from users. Furthermore, installation and maintenance of the sensors usually entail high costs which hinders a real time HAR system to be scalable [73].

Advantages of using environmental devices strictly depend on the context and on the application. The next paragraph aims at presenting typologies, strengths and weaknesses of wearable devices and sensors.

2.1.2 Wearable Devices and Sensors

Wearable devices encompass all accessories attached to the person's body or clothing incorporating computer technologies, such as smart clothing, and ear-worn devices [53]. They enable to capture attributes of interest as motion, location, temperature and ECG, among others.

Wearable devices and HAR are very interconnected. Indeed, over the last years, wearables gained the attention of HAR because of many reasons. Despite environmental, wearable devices are possible to carry out-door and are much cheaper. Recently, most of wearable devices have spread among the population, and, consequently, a tremendous increase in wearable's use in many application's areas has been reported [122]. Wearables are used for security, wellness, medical, sport, and many others.

Smartphones and smartwatches are the most used wearable devices among the population. In particular, the smartphone is one of the most used devices in people's daily lives and it have been stated that it is the first thing people reach for after waking up in the morning [33, 97].

Smartphone's pervasiveness over last years, is due mostly because it provides the opportunity to connect with people, to play games, to read emails, and, in general, to achieve almost all online services that a user needs. In particular, their high diffusion is a crucial aspect because the more the users, the more data availability. The more data availability, the more information and the more the possibility to create robust models. A the same time, smartphones are preferable over other wearables because a huge amount of sensors and softwares are already installed and permit to acquire many kind of data, potentially, all day long. Figure 2.2 shows all embedded sensors in a smartphone.

The choice of the sensors plays an important role for the activity recognition performances for which abundant literature has been written[115].

The aim of this section is to describe the most used sensors for HAR tasks and classification.

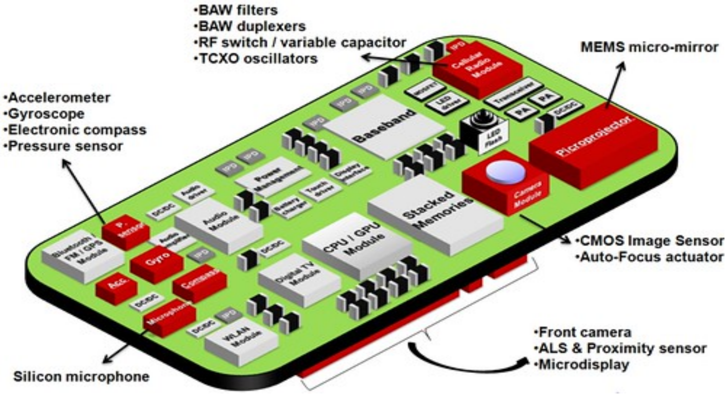


Figure 2.2: Smartphone's embedded sensors.

Accelerometer. The accelerometer is an electromechanical sensor dedicated to capture the rate of change of the velocity of an object over a time laps, i.e. the acceleration. It is composed of many other sensors, including some microscopic crystal structures that become stressed due to accelerative forces. The accelerometer interprets the voltage coming from the crystals to understand how fast the device is moving and which direction it is pointing in. A smartphone records three dimension acceleration, which join the reference devices axes. Thus, a trivariate time series is produced. The measuring unit is meters over second squared (m/s2) or g - forces.

Gyroscope. The gyroscope measures three-axial angular velocity. Its unit is measured in degrees over second (degrees/s). Although accelerometer still remains the most used sensor for HAR, many studies have exploited also gyroscope for activity [113]. The reason is two folds: the addition of more information about the device mouvements, and the possibility to infer the device's position..

Magnetometer. A magnetometer measures the change of a magnetic field at a particular location. The measurement units are Tesla (T), and is usually recorded on the three axes.

Global Positioning System. GPS units inside phones gets a ping from a satellite in space and based on angles intersection they calculate the device location. It is normally used for sport tracking and it is mostly combined with the accelerometer.

For the sake of completeness, we briefly mention other state-of-the-art sensors. In [91] the barometer have been used. Its functionality is related to vertical activities, such as ascending and descending stairs. Pressure is mentioned in [32, 48], and biometric sensors, as electromyography(EMG) for fine-grained motions has been used in [150]. Electrocardiography (ECG) in [81].

Accelerometer is the most popular sensor in HAR because it measures the directional movement of a subject's motion status over time [13, 76, 109, 140, 90, 3]. Nevertheless, it struggles to resolve lateral orientation or tilt, and to find out the location of the user, which are precious information for activity recognition. For these reasons, some sensor's combinations have been proposed as valid solution in HAR. In most of the cases accelerometer and gyroscope are combined [4, 60, 58].

Authors in [113] demonstrated that gyroscope based classification achieves better results than accelerometer for specific activities, i.e. walking downstairs and upstairs. Furthermore, as afore mentioned, gyroscope data permit to infer device position that drastically impacts recognition performances [127, 17].

Other studies combined accelerometer and magnetometer simultaneously [118], acceleration and gyroscope with magnetometer [149, 119], accelerometer with microphone and GPS [72], and many other combinations [77].

2.1.3 Conclusions

The recent years have been characterized from drastic software and hardware changes. On the hardware side, abundant number of low cost devices and sensors have been developed.

For instance, smartphones and smartwatches are, nowadays, very cheap and equipped by many different micro-sensors.

On the software side, new devices are able to store huge amount of data, interconnect many devices at once and share information. Human activity recognition exploited these new technologies to acquire, store and analyze data. In particular, two types of devices are used in HAR: environmental and wearable devices. Environmental devices are installed in the environment or placed on objects, such as camera and tags. Wearable devices are placed on the user's body, e.g. smartphone, smartwatches, and ad-hoc devices. In general, wearables are suitable for out-door context, e.g. for sport tracking or activity recognition, while environmental sensors are preferable for complex activities related to the in-door context. However, the choice between environmental or wearable devices is a difficult task and should be done depending on several factors, such as the application domain, the activities of interest, i.e. complex activity or basic activity, the general context, e.g. if out-door or in-door.

Nevertheless, in the recent years, wearables gained a large attention in the human activity research community mostly because of their pervasiveness among the population. According to [33], smartphones are the most used devices in the population's daily life. HAR developed many methods and instruments to manage data based on wearables devices and on smartphones. Among sensors, 3-axis acceleration is the most exploited sensor in HAR. Normally used alone, it is also combined with the gyroscope.

The possibility to monitor people daily activities, risky activity or changes in behavior, e.g. falls or disease's development, habits, with a simple smartphone is very attractive and actual. For this reason, in this work, we concentrated on data recorded by smartphones for the classification. In particular, we consider data acquired from 3-axis accelerometer and gyroscope embedded in a smartphone.

2.2 Data Preprocessing

In Section 2.1, the most suitable devices for human activity recognition are presented and describes. In the recent years, the spread of wearable devices has lead to a huge availability of physical activity data. Smartphones and smartwatches are become more and more pervasive and ubiquity in our everyday life. This high diffusion and portability of wearable devices has permitted scientists to easy produce plenty of labeled raw data for human activity recognition.

Several public datasets are open to the HAR community and are easy accessible on the web, see for instance the UC Irvine Machine Learning Repository [14]. In this study we discuss and analyze smartphone-based datasets. We believe indeed that the pervasiveness and portability of smartphones make this device the most powerful among wearables, in terms of capability of monitoring user's daily life.

Usually, public datasets collected by research groups and repositories provide raw data, which need to be processed and structured before to be considered valid input data for analysis and machine learning engine.

In subsection 2.2.1 the available public benchmark datasets are presented, while in subsection 2.2.2 state-of-the-art approaches for data segmentation and feature extraction are described.

2.2.1 Benchmark Datasets

In the last decade, a huge amount of public datasets for HAR has been produced. As discussed in Section 2.1, activity recognition classifiers exploit data collected from environment, object and wearable devices. According to the goal of this work, we selected datasets which collect inertial sensor data of Activity of Daily Living (ADLs) recorded from smartphones.

In Table 2.1, we show the main characteristics of the most exploited datasets in the state-of-the-art. Datasets which combine smartphones and IMUs or smartphones and smartwatches are also considered, see datasets D03, D010, D11, and D16.
In column *# Activities* the number of ADLs is shown. Usually, from about 6 to 10 ADLs are recorded and in some cases, both ADLs and Falls data are considered, as in datasets D08, D09, D11. We decided to do not discard Falls data when collected with ADLs and include them into our analysis.

The column *# Subjects* reports the number of the subjects which performed the activities. Considering a restricted number of subjects in the analysis does not just impact the quality and robustness of the classification, but also the ability to evaluate the consistency of results across subjects [83]. In others words, the number of the subjects includes in the training set of the algorithm is crucial in terms of generalization capability of the model to classify a new unseen instance.

Nevertheless, the different between people, also called *population diversity*, could lead to poor classification, as largely discussed in [72]. Unfortunately, most of the datasets are limited in terms of subject numerousness. To overcome this issues, recently, several HAR research groups implemented strategies for merging datasets [46, 117]. Other techniques, such as transfert learning and personalization, have been investigated for robustness of results [104, 41, 82].

Column *Devices* reports typologies and number of devices that have been used to collect data. In particular, datasets D03, D04, D05, D06, D11, D12 collected data from several wearable devices at the same time, which is due to the following reasons. First, the device position influences the performance of the classification. Several works investigated which position leads to the best classification [69, 113]. Furthermore, it is also challenging to investigate device's fusion, which has a not negligible positive effects on the classification performances and reflects realistic situation where users use many smart devices at once [128, 65, 5, 90].

Position-aware and position-unaware scenario have been presented in [113]. In position-aware scenario the recognition accuracy on different positions individually is evaluated, while in position-unaware scenario the classification performance of the combination of devices positions is measured. It is shown that the latter approach highly improves the classification performance for some activities, such as walking, walking upstairs and walking downstairs. In [3] they exploited deep learning technique for classification and demonstrated its capability to produce an effective position-independent HAR. In this study, we do not focus on a specific position of smartphone because we concentrate mostly on subject-related classification effects.

ID	Dataset	# Activity	# Subject	Devices (#)	Sensors	Sampling Rate (Hz)	Metadata	Reference
D01	UCI HAR	6 ADL	30	SP(1)	A,G	50	no	[7]
D02	Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set	6 ADL	30	SP(1)	A,G	50	no	[7]

D03	HHAR	6 ADL	9	SP(8)
				SW(4)
					A,G	H	no	[123]

D04	Physical Activity Recognition Dataset Using Smartphone Sensors	6 ADL	4	SP(4)	A,G,M	50	no	[115]
D05	Sensors activity dataset	7 ADL	10	SP(5)	AG, M, LA,	50	no	[113]
D06	Complex Human Activities Dataset	13 ADL	10	SP(2)	A, G, LA	50	no	[114]
D07	Motions Sense	6 ADL	24	SP(1)	A,G, AT	50	Gender, Age, Height, Weight	[86]
D08	MobiAct	11 ADL, 4F	67	SP(1)	A, G, OR	87	Gender, Age, Height, Weight	[135]
D09	UniMiB-SHAR	9 ADL, 8 F	30	SP(1)	A	50	Gender, Age, Height, Weight	[88]
D10	UMAFall	12 ADL, 3 F	19	SP(1), IMUs(4)	A, G, M	200,20	Gender, Age, Height, Weight	[29]
D11	Real World	8 ADL	15	SP(6), SW(1)	A, G, GPS, L, M, S	50	Gender, Age, Height, Weight	[127]
D12	WISDM	6 ADL	29	SP (1)	A	20	no	[70]
D13	Smartphone Dataset for HAR in Ambient Assisted Living (AAL) Data Set	6 ADL	30	SP (1)	A, G	50	no	[7]
D14	Daily Activity Dataset	5 ADL	8	SP (1)	A	40	no	[120]
D15	HASC2010	6 ADL	96	SP(1)	A	from 10 to 100	Gender, Height,Weight, Shoes, Floor, Place	[67]
D16	Extrasensory Dataset	7 ADL + 109 Specific Activities	60	SP(1), SW	A, G, M, CO, LO, S, SM, ST	40,25	no	[132]

Table 2.1: Public HAR dataset collection inertial signals recorded from smartphone. (ADL = Activity of daily living, F = Falls; A = Accelerometer, LA = Linear Acceleration Sensor G = Gyroscope, M = Magnetometer, AT = attitude, OR = orientation, L = light, S = sound, SM = sound magnitude GPS = Global Positioning System,CO= compass, LO = location, ST = phone state, H = highest frequency as possible, SP = smartphone, SW = smartwatch, IMU = inertial measurement unit

Column *Sensors* lists the sensors exploited in data collection. Tri-axial acceleration data (A) is the most exploited inertial sensor among the literature[73]. Most recent studies exploiting machine learning and deep learning techniques largely used acceleration sensor [140, 104, 5]. Datasets D9, D14, and D15 even collected just acceleration data. Acceleration is very popular because it directly captures the subject's physiology motion status and its low energy consumption [62].

Acceleration has been combined with other sensors, such as gyroscope, magnetometer, GPS, and biosensors with the aim of improving activity classification performance. In general, data captured from several sensors carry additional informations about the activity and about the device settings. For instance, information derived from gyroscope is used to maintain reference direction in the motion system and permit to determine mobile orientation [125, 4]. Performances comparison between gyroscope, acceleration and their combination for human activity recognition have been explored in many studies [43, 113].

Authors in [43] showed that accelerometer is more performing than the gyroscope and their combination leads to an overall improvement of about 10%. In [113], authors state that in situations where accelerator and gyroscope individually perform with low accuracies, their combination improved the overall performance, while when one of the sensors performs with higher accuracy, the performance doesn't improve combining sensors.

In column *Sampling Rate* is shown the frequency the data are acquired. As stated above, data acquisition phase is responsable to take into account how data are recorded. The sampling rate has to be high enough to capture most significant behavior of data. In HAR most used sampling rate is 50Hz.

Column *Metadata* list characteristics regarding the subjects which perform the activities. In D07-11, D15 physical characteristics are annotated. In D15 environmental characteristics have been also stored, such as the kind of shoes wear, floor characteristics and place where activities have been preformed. As discussed in section 2.3 there are benefits in using metadata on the activity recognition performance. Indeed, metadata are precious additional information, which help to overcome population diversity issue. For experiments on datasets D07, D08, D09, using metadata, see section 3.

Figure 2.3 shows datasets distribution over last decade.

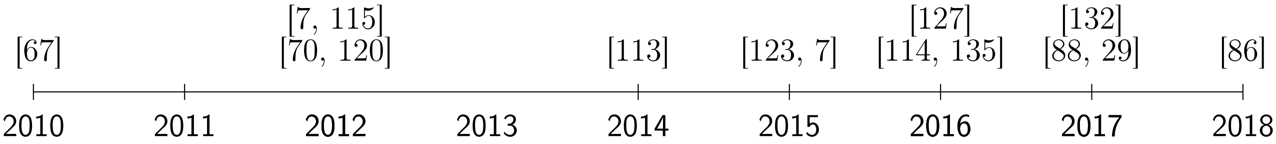


Figure 2.3: Distribution of HAR Datasets on wearable devices over last decade.

2.2.2 From Raw Data to Input Data

The community of HAR have collected and published plenty of datasets. Generally, data collected in datasets are still considered raw data because they are not ready to use as input data for analysis. Depending on the analysis, data have to be transformed and prepared. For machine learning techniques, data should pass the Activity Recognition Process (ARP), which is composed in four phases shown in Figure 2.1.

In the following sections, more details about *Data Segmentation*, *Features Extraction*, and *Classification* phases have been discussed. Data acquisition and pre-proccessing phases haven't been considered because out of our study context.

2.2.2.1 Data Segmentation

Data segmentation is the process that partitions raw signals into smaller data segments, also called windows. It can be classified into three categories, namely activity-defined windows, event-defined windows and sliding windows. Initial and end point of the activity-defined windows are selected by detecting patterns of the activity changes.

Event-defined windowing procedure consists of creating a window around a detected event. In some studies it is also mentioned as windows around peak [88]. Sliding windows is the most widely employed segmentation technique in activity recognition, especially for periodic and static activities [11]. It consists of splitting data into windows of fixed size, without gap between two consecutive windows, and, in case, overlapped, as shown in Figure 2.5.

Data segmentation is essential to overcome some limitations related to many acquisition and pre-proccessing aspects. First, the data sampling: data recorded from different subjects may present different lengths in time which is generally a limit for the classification process. Second, the time consumption: multidimensional data can lead to a very high computational time consumption. Split data into smaller segments helps the algorithm to face with high volume of data. Third, it helps the computation of the features extraction procedure in terms of more simplicity and lower time consuming.

Data segmentation phase is crucial for the analysis because it determines the the input data's structure which is critical for the classification performance and for availability of results.

In particular, the choice of the window size is determinant for the accuracy of the classification [30]. The choice of the window size is not trivial. It should be large enough to guarantee to contain at least one cycle of an activity and to differentiate similar movements, and, at the same time, incrementing it too much doesn't improve necessarily the performance.

In [113], authors showed that 2s is enough for recognizing basic physical activities.

Figure 2.4 shows distribution of windows size among state-of-the-art studies we considered. In the 56% of the cases, windows size is less or equal than 3 seconds. Window size between 3 and 5 seconds is considered in the 17% of the cases, while the 13% chose it between 5 and 10 seconds. In very few case, the 4%, windows size exceeds 10 seconds. The impact of windows size on the classification performance still remains a challenging task for HAR community and continues to be largely studied in the literature [113, 65, 11].

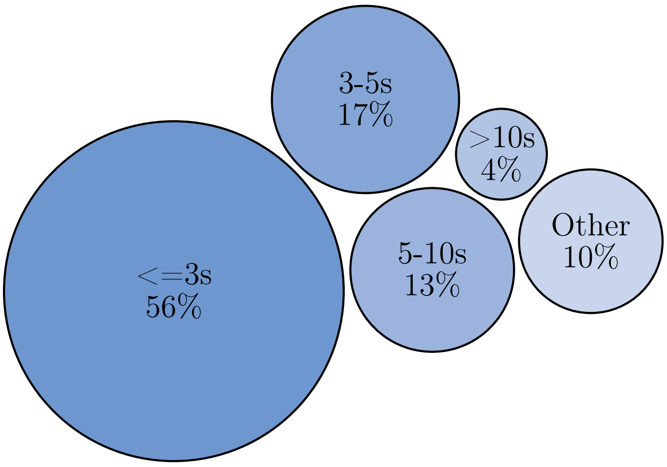


Figure 2.4: State-of-the-art Sliding Window's Size.

Another parameter to be chosen is the percentage of overlap. Sliding windows are often overlapped which means that a percentage of a window is repeated in the subsequent window. This leads to two main advantages: it avoids noise due to the truncation of data during the windowing process and increases the performance by increasing the data points number. Generally, the higher the number of data points, the higher the classification performance.

For these reasons, overlapped sliding windows is the most common choice in the literature. Figure 2.6 shows the distribution of the percentage overlap instate-of-the-art. In more than 50% of the papers we selected, the 50% of overlap has been chosen.In some cases [28, 140, 4, 68] no-overlap has been chosen, for instance to allow a fast response in real time and for detection of short duration movements.

Given the advantages to use overlapped windows, our analysis is based on this approach.

In the following details about data segmentation of this work are exposed. Let's be x, y, z the 3-axis acceleration values. After data segmentation phase, data are organized in vectors $\mathbf{v}_i$ as follows:

$$\mathbf{v}_i = \big(\underbrace{x_1,\ x_2, x_3\ \dots x_n,}_{x-dimension}\ \underbrace{y_1,\ y_2, y_3,\ \dots\ y_n,}_{y-dimension}\ \underbrace{z_1,\ z_2,\ z_3,\ \dots\ z_n}_{z-dimension}\big)$$

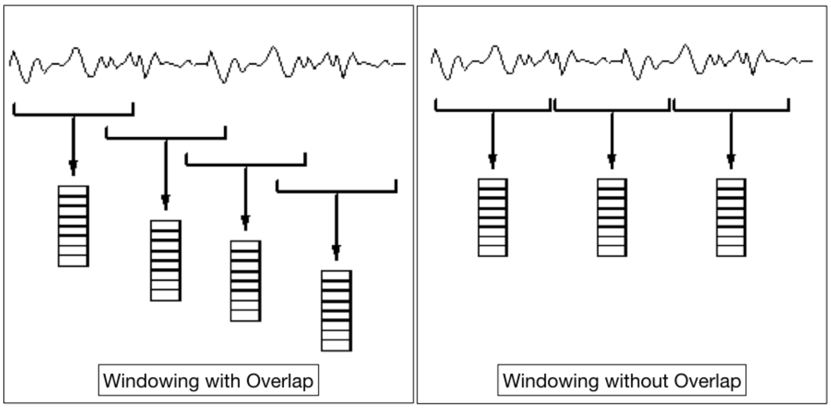


Figure 2.5: Sliding windows with and without Overlap

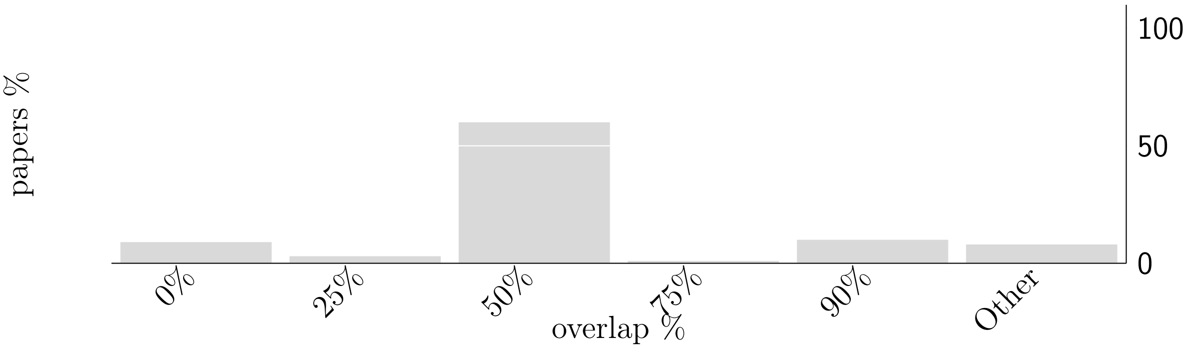


Figure 2.6: Overlap % used among state-of-the-art.

where $\mathbf{v}_i$ is a $1 \times (n \times k)$ vector, which represents the i-th window. k refers to the number of the sensor dimension, for instance if 3-axial acceleration is recorded $k = 3$. The number n is the total length of the windows which depends on two factors: the size of the widows, normally in seconds, and the *sampling rate*. The sampling rate is define as the number of the data points recorded in a second and expressed in Hertz. For instance, if the frequency rate of sampling is equal to 50Hz, it means that 50 values per second are recorded. This parameter is normally set during the acquisition phase. Modern device can be set with a specific sample rate value. The choice of the sampling rate is crucial.

In the literature different sampling rates have been chosen. For instance, in [88] the sample rate is set at 50Hz, in [68] at 45Hz, and from 30 to 32Hz in [4]. Although the choice is not unanimous in the literature, 50Hz define a suitable sampling rate that properly permits to model human activities [101]. In this study we considered dataset at their own sample rate, larger or equal to 50Hz, while in some case we use a linear interpolation procedure to homogeneize all sampling rate at 50Hz.

2.2.2.2 Features Extraction

Theoretical analysis and experimental studies indicate that many algorithms scale poorly in domains with large number of irrelevant and\or redundant data.

In literature it is shown that using a set of features instead of raw data improves the classification accuracy [40]. Furthermore, features extraction reduces the data dimensionality while extracting the most important peculiarity of the signal by abstracting each data segments into an high-level representation of the segment.

From a mathematical point of view, features extraction is defined as a process that extracts a set of new features from the original data segment through some functional mapping [80]. For instance, let be $\mathbf{x} = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^n$ a segment of data, an extracted feature f_i is given by

$$f_i = g_i(x_1, x_2, \dots, x_n) \qquad \text{for } i = 1, \dots, m$$

where $g_i : \mathbb{R}^n \to \mathbb{R}$ is a map. Features space is of dimension $m \leq n$, which means that features extraction reduces raw data space dimension, in general.

In the classification context, the choice of g_i is crucial. In fact, in the recognition process, g has to be chosen such that the original data are mapped in separated regions of the features space. In other words, the researcher assumes that in the feature space data separate better than in the original space. The accuracy of activity recognition approaches dramatically depends on the choice of the features [30]. In the literature, the way features g_i are extracted is divided into two main categories, *hand crafted features* and *learned features*.

Hand-crafted features are the most used features in HAR [82, 70, 5]. The term "hand-crafted" reminds to the fact that the features are selected from an expert using heuristics. Hand-crafted features, in turn, are generally split in *time domain* and on *frequency domain* features. The signal domain is changed from the time to the frequency based on the Fourier transformation. In Table 2.2 most used time domain and frequency domain features are listed and described.

Low computational complexity and calculation simplicity make hand-crafted features still a good practice for activity recognition. Nevertheless, they present many disadvantages, such as an high dependency on the sensor choice and the dependency on the expert knowledge. Hence, a different set of features need to be defined for each different type of input data i.e. accelerometer, gyroscope, time-domain and frequency domain. In addition, hand-crafted features highly depend on expert prior knowledges and manual data investigation and it is still not always clear which features are likely to work best.

Time Domain Features

Name Features	Formula	Description
Minimum	$\min_{j=1,\dots n}(x_j)$	The mean value of a given segment in each dimension
Maximum	$\max_{j=1,\dots n}(x_j)$	Maximum value of a given segment in each dimension
Mean	$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$	Minimum value of a given segment in each dimension
Median	$Me = x_{0.5} : F(x_{0.5}) \leq 0.5$	Median of a given segment in each dimension
Standard Deviation	$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$	Standard Deviation of a given segment in each dimension
Variance	$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$	Variance of a given segment in each dimension
Interquartile Difference	$ID = x_{0.75} - x_{0.25}$	Difference between third and first quartile of a given segment in each dimension
Skewness	$skw = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{s^3}$	Skewness value of a given segment in each dimension
Kurtosis	$kurt = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{s^4}$	Kurtosis value of a given segment in each dimension
Root Mean Square	$rms = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$	Root mean square value of a given segment in each dimension
Total Sum	$ts = \sum_{i=1}^{n} x_i$	Total sum value of a given segment in each dimension
Range	$R = max - min$	Range of a given segment in each dimension
Mean of Peak's Distance	$m_p = \frac{1}{s^2}\sum_{j=1}^{s}\sum_{i=1}^{s} d(p_i, p_j)$	Mean of distance between peaks of a given segment in each dimension
Fourth central moment	$m_4 = \frac{1}{n}\sum_{j=1}^{n}(x - \bar{x})^4$	Fourth central moment of a given segment in each dimension
Fifth central moment	$m_5 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^5$	Fifth central moment of a given segment in each dimension

Frequency Domain Features

Name Features	Formula	Description
Entropy	$H(x) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$	Normalized information entropy of the discrete FFT components
Sum of the spectral power components	$SP = \frac{1}{n}\sum_{j=1}^{f} \lvert FFT_j \rvert^2$	Mean of the square of absolute value of FFT
Mean of the spectral components	$\mu_f = \frac{1}{n}\sum_{j=1}^{n} FFT_j$	Mean of FFT distribution
Median of the spectral components	$Me_f = FFT_{0.5} : F(FFT_{0.5}) = 0.5$	Median of FFT distribution
First Cepstral Coefficient	$c(1) = \mathcal{F}^{-1}\{\log \lvert FFT(f) \rvert\}$	First coefficient of the cepstrum transofrmation

Table 2.2: Hand-crafted Features in Time Domain and Frequency Domain.$FFT(f)$ is the Fourier Transformation of the signal f.

It is a common practice to chose the features through empirical evaluation of different combinations of features or with the aid of feature selection algorithms [108].

Learned Features: the goal of feature learning is to automatically discover meaningful representations of raw data to be analyzed [98]. According to [71], main features learning methods from sensor data are the following:

- *Codebooks* [137, 111] consider each sensor data window as a sequence, from which subsequences are extracted and grouped into clusters. Each cluster centre is a codeword. Then, each sequence is encoded using a bag-of-words approach using codewords as features.
- *Principal Component Analysis (PCA)* [2], is a multivariate technique, commonly used for dimensionality reduction. The main goal of PCA is the extraction of a set of orthogonal features, called principal component, which are linear combination of the original data and such as the variance extracted from the data is maximal. It is also used for features selection.
- *Deep Learning*, uses Neural Networks engine to learn patterns from data. Neural Networks are composed from a set of layers. In each layer, the input data are transformed through combinations of filters and topological maps. The output of each layer becomes the input of the following layer and so on. At the end of this procedure, the result is a set of features more or less abstract depending on the number of layers. The more the number of layers is high, the more the features are abstract. These features can be used for classification. Different deep learning methods for features extraction have been used for time series analysis [44].

Features learning techniques avoid the issue to create and select manually the features. Recently, promising results are more and more lead the research community to exploit learned feature in their analysis.

2.2.3 Conclusions

HAR community has published plenty of datasets for human activity recognition based on inertial sensors embedded in smartphones, but data have to be transformed before to be ready for the classification. Activity Recognition Process (ARP) is responsible to transform raw data into input data through four phases: data acquisition, data preprocessing, data segmentation, feature extraction, classification. Usually, public datasets have already passed data acquisition and data preprocessing phases.

In this work, the analysis have been done on public datasets and for this reason, only data segmentation, features extraction and classification have been discussed in this section. All phases are critical for the performance of the classification and each of them depends on many parameters that have to be choose. In this work, we based on state-of-the-art best results and benchmarks to implement our ARP. We selected three datasets to implement the personalized model for machine learning and deep learning, namely Motion Sense (D7), MobiAct (D8), and UniMiB-SHAR (D9), because they encompass physical characteristics of the subjects. For other experiments we add UCI HAR (D1), as it is the most used dataset in the literature.

2.3 Classification and Evaluation: Methods for Automatic HAR

Over the last years, hardware and software development has increased wearable devices capability to face with complex applications and tasks. For instance, smartphones are, nowadays, able to acquire, store, share and elaborate huge amount of data in a very short time. As consequence of this technological development, new instruments related to the data availability, data processing, data analysis are born. The capability of a simple smartphone to meet some complex tasks, e. g. steps count, life style monitoring, is the results of a very recent scientific changes regarding methods and techniques.

In general, more traditional data analysis methods, based on model-driven paradigms, have been largely substituted by more flexible techniques, developed during the recent years, based on data-driven paradigms. In a few words, the main difference between these two approaches is given by the a priori assumption about the relationship between independent and response variables.

The strength and the success of data-driven approaches are due to their capability to manage and to analyze large amount of variables that characterize a phenomenon without assuming any a-priori relation between the independent and response variables. From a certain point of view, this flexibility can be also a weakness because the lack of a well-known relation also can be interpreted as a lack of cause-effect knowledge. In model-driven approaches, in contrast, cause-effect relation is known by definition, but their loose in performance in estimating high-dimensionality relations. In activity recognition context, model-driven approaches are not powerful and data-driven approaches are preferred.

Among data-driven algorithms, Artificial Intelligence (AI) have produced very promising results over the last years and have been largely used for data analysis, for information extraction and for classification tasks. AI emulates the human behavior, reasoning and learning. AI algorithms encompasses machine learning which, in turns, encompasses deep learning methods. Machine learning uses statistical exploration techniques to enable the machine to learn and improve with the experiences without being explicitly programmed. Deep learning emulates human neural system to analyze and extract features from data. Figure 2.7 shows the relationship between AI, machine learning and deep learning. In this study, we concentered on machine learning and deep learning algorithms.

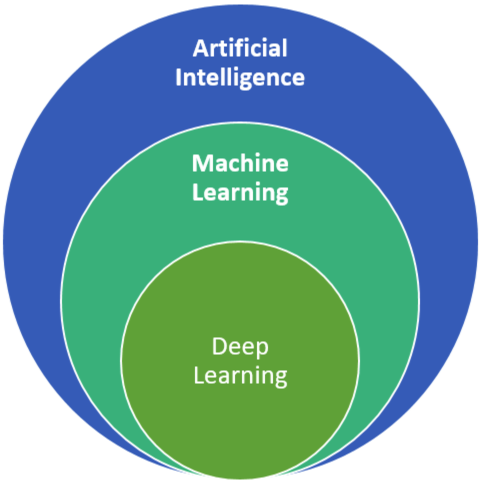


Figure 2.7: Artificial Intelligence, Machine Learning and Deep Learning.

The choice of the classification's algorithm drastically influences the classification performance, but up to now, there is no evidence of a best classifier and its choice still remains a challenging task for the HAR community.

In particular, machine learning and deep learning methods struggle to achieve good performances for new unseen users. This lost of performance is mostly caused by the subjects variabilities, also called *population diversity* [72] which is related of the natural users heterogeneity in terms of data.

Before entering into details, traditional state-of-the-art machine learning and deep learning techniques are presented in Section 2.3.1. In Section 2.3.2 we investigated and discussed about personalized machine learning and deep learning techniques as solutions to overcome the population diversity issue.

2.3.1 Traditional Learning Methods

As mentioned above, Artificial Intelligence (AI) algorithms base on the emulation of the human learning. According to [148], the word *learning* refers to a process to acquire knowledge or skill about a thing. A thing can always be viewed as a system, and the general architecture of the knowledge of the thing follows the *FCBPSS* architecture, in which F: function that refers to the role a particular structure plays in a particular context; C: context that refers to the environment as well as pre-condition and post-condition surrounding a structure; B: behavior that refers to causal relationships among states of a structure; P: principle that refers to the knowledge that governs a behavior of a structure; S: state that describes the property or character of a structure; S: structure that represents elements or components of the system or thing along with their connections [79].

Machine learning and deep learning both refer to the word *learning* and, indeed, they are implemented so that they emulate the human capability of learning. In the following, more details about traditional learning methods are highlighted, in particular we concentrate on machine and deep learning.

Machine learning techniques used in HAR are mostly subdivided into *supervised* and *unsupervised* approaches. Supervised machine learning encompasses all techniques referred to labeled data. Unsupervised machine learning are techniques which based on data devoid of labels.

In terms of classification algorithm structure, the subdivision between supervised and unsupervised is crucial. Let $\mathbf{x}$ and $\mathbf{y}$ be, respectively, a set of data and their corresponding labels. A *classification task* is a procedure which goal is to predict the value of the label $\hat{\mathbf{y}}$ from the data input $\mathbf{x}$. In other terms, assuming that there exists a linear or non-linear relation f between $\mathbf{x}$ and $\mathbf{y}$, the goal of the classification is to find f such as the prediction's error, i.e. the distance between $\mathbf{y}$ and $\hat{\mathbf{y}}$, is minimal. In supervised machine learning, data and corresponding labels are known and the algorithm learns f by iterating a procedure until the global minimum of a loss function is reached. The loss function is again a measure about the prediction's error, estimated by the difference between $\mathbf{y}$ and $\hat{\mathbf{y}}$. The optimization procedure, i.e. finding the loss global minimum, is computed on the *training dataset*, which is a subset of the whole dataset. Once the minimum is achieved, the model is ready to be tested on the *test dataset*. The *algorithm performance* measure the model's capability to classify new instances, see Section2.2 for more details about the performance measures.

In unsupervised approaches, the labels $\mathbf{y}$ are unknown and the evaluation of the algorithm goodness bases on statistical indices, such as the variance or the entropy. Consequently, the choice between supervised or unsupervised methods determines how the relation f between $\mathbf{x}$ and $\mathbf{y}$ is learnt. Since a human activity recognition system should return a label such as walking, sitting, running, etc., most of HAR systems work in a supervised fashion. Indeed, it might be very hard to discriminate activities in a completely unsupervised context [73].

In the following paragraph, *traditional machine learning* is discussed. The term *traditional* refers to the standard machine learning and deep learning methods. In Figure2.8 the distribution of traditional machine learning and deep learning algorithms used for human activity recognition is shown. This distribution is the result of the state-of-the-art study we mentioned at the beginning of this chapter. We can notice that machine learning as well as deep learning techniques are largely exploited in HAR. In the following paragraphs, we will describe the most used techniques in HAR with the related literature.

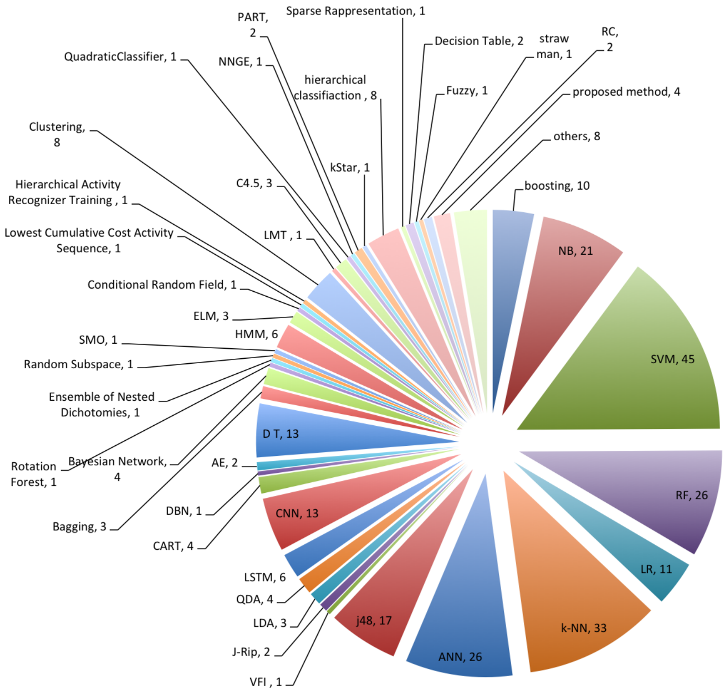


Figure 2.8: Traditional Machine Learning and Deep Learning classifiers distribution.

2.3.1.1 Traditional Machine Learning

Machine learning techniques have been largely used for activity recognition tasks. More and more sophisticated methods have been developed to face with the complexity related to activity recognition tasks. In this section we describe traditional machine learning algorithms that have been mostly exploited for human activity recognition, according to the Figure 2.8.

Support Vector Machines (SVM) belongs to *domain transform* algorithms. It implements the following idea: it is assumed that the input data $\mathbf{x}$ are not linearly separable with respect to the classes $\mathbf{y}$ in the data space, but there exists an higher dimensional space where the linearity is achieved. Once data are mapped into this space a linear decision surface (or hyperplane) is constructed and used as recognition model. Thus, guided from the data, the algorithm searches for the optimal decision surface by minimizing the error function. The projection of the optimal decision surface into the original space marks the areas belonging to a specific class which are used for the classification [34]. The transformation of the original space into a higher dimensional space is made through a *kernel* which is defined as a linear or non linear combination of the data, e.g. polynnomial kernel, sigmoid kernel and radial basis function (RBF) kernel, see Table 2.3. Originally, SVM have been implemented as two-class classifier. The computation of the multi-class SVM bases on two strategies: *one-versus-all* where one class is labeled with 0 and the other classes as 1, and *one-versus-one* where the classification is made between two class at a time [31]. Among HAR classifiers, SVM is the most popular one[4, 58, 6, 134, 143, 64, 15, 31].

Kernel	*Linear*	*Polynnomial*	*RBF*
Formula	$\mathbf{x}_i^T\mathbf{x}_j$	$(\mathbf{x}_i^T\mathbf{x}_j + c)^d$	$exp\left(\frac{\vert\vert\mathbf{x}_i-\mathbf{x}_j\vert\vert^2}{2\sigma^2}\right)$

Table 2.3: Kernel in Support Vector Machines.

***k*-Nearest Neighbors (*k*-NN)** is a particular case of *instance based* methods. The nearest neighbour algorithm compares each new instance with existing ones using a distance metric, see Table 2.4, and the closest existing instance is used to assign the class to the new one. This is the simplest case where $k = 1$. If $k > 1$, the majority class of the closest k neighbors is assigned to the new instance [141]. It is a very simple algorithm and belongs to the lazy algorithms. Lazy algorithms have no parameters to learnt from the training phase [4, 6, 134, 114, 31]. k-NN depends only on the number k of the nearest neighbors.

Distance	*Eucleadin*	*City Block*	*Chebychev*
Formula	$\sqrt{\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{x}_j)^2}$	$\sum_{i=1}^{n} \vert\mathbf{x}_i - \mathbf{x}_j\vert$	$\max_{i=1\dots n} \vert\mathbf{x}_i - \mathbf{x}_j\vert$
Distance	*Cosine*	*Correlation*	*Mahalnobis*
Formula	$1 - \frac{\mathbf{x}_i\mathbf{x}_j^T}{\sqrt{(\mathbf{x}_i\mathbf{x}_i^T)(\mathbf{x}_j\mathbf{x}_j^T)}}$	$1 - \frac{(\mathbf{x}_i-\bar{\mathbf{x}}_i)(\mathbf{x}_j-\bar{\mathbf{x}}_j)^T}{\sqrt{(\mathbf{x}_i-\bar{\mathbf{x}}_i)(\mathbf{x}_i-\bar{\mathbf{x}}_i)^T}\sqrt{(\mathbf{x}_j-\bar{\mathbf{x}}_j)(\mathbf{x}_j-\bar{\mathbf{x}}_j)^T}}$	$\sqrt{(\mathbf{x}_i - \mathbf{x}_j)C^{-1}(\mathbf{x}_i - \mathbf{x}_j)^T}$ wher C is the covariance matrix

Table 2.4: Distance metrics in k-nearest neighbor.

J48 and C4.5 belong to *decision tree* algorithms. Decision tree algorithms build a hierarchical model in which input data are mapped from the root to leafs through branches. the path between the root and a leaf is a classification rule [73]. Sometimes, trees length has to be modified and growing and pruning algorithms are used. The construction of a tree involves determining split criterion, stopping criterion and class assignment rule [103].

J48 and C4.5 are the most used decision tree in HAR [140, 90, 134, 114].

Random Forest (RF) is a classifier consisting of a collection of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k), k = 1, ...\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\mathbf{x}$ [24]. Random Forest generally achieves high performance with high dimensional data by increasing the number of trees [65, 140, 149, 99, 12, 15, 31].

Naive Bayes (NB) belongs to *bayesian* methods which prediction of new instances is based on the estimation of the a posterior probability as a product of the likelihood, which is a conditional probability estimated on the training set given the class, and a prior probability. In Naive Bayes, data are assumed independent given the class values. Thus, given y be a certain class and $\mathbf{x}_i ... \mathbf{x}_n$ the data, Naive Bayes classifier based on the baysian rules and the likelihood splits in the product of the conditional probabilities given the class

$$P(y|\mathbf{x}_1...\mathbf{x}_n) = \frac{P(y)P(\mathbf{x}_1, ...\mathbf{x}_n|y)}{P(\mathbf{x}_1, ...\mathbf{x}_n)} = \frac{P(y)\prod_{i=1}^{n} P(\mathbf{x}_i|y)}{P(\mathbf{x}_1, ...\mathbf{x}_n)}$$

Decision rules is the maximum a postriori (MAP) given by

$$\arg\max_y P(y|\mathbf{x}_1...\mathbf{x}_n) = \arg\max_y \prod_{i=1}^{n} P(\mathbf{x}_i|y)$$

Naive Bayes has been applied in activity recognition because its simple assumption on the likelihood, which is usually violated in practice [8, 140, 134, 114]

Adaboost is part of the *classifier ensembles*. Classifier ensembles encompasses all algorithms that combine different classifiers together. The combination between the classifiers is meant in two ways: either using the same classifiers with different parameter's settings, e.g. random forest with different lengths, or using different classifiers, e.g. random forest, support vector machines and k-NN together. The ensemble classifiers encompass *bagging*, *stacking*, *blending*, and *boosting*. In bagging, n samplings are generated from training set and a model is created on each. The final output is a combination of each model's prediction. Normally, either the average or a quantile is used. In stacking, the whole training dataset is given to the multiple classifiers which are trained using the k-fold-cross-validation. After training they are combined for final prediction. In blending, the same procedure as staking is performed but instead of the cross-validation, the dataset is divided into training and validation. Finally, in boosting, the final classifier is composed of a weighted combination of models. The weights are initially equal for each model and are iteratively updated based on the models performance, e.g. Adaboost [73, 72, 93].

Human activity recognition is mainly carried out with the support of machine learning techniques. Machine learning techniques are preferable to deep learning techniques because of many reasons. They are very performant even with low amount of labeled data and are low time-consumption methods. Nevertheless, machine learning techniques remain highly expertise dependent algorithms. Input data feeding machine learning algorithms are normally features, a processed version of the data. Features permit to reduce data dimensionality and computational time. However, features are handcrafted and are expert knowledge and tasks dependent. Furthermore, engineered features cannot represent salient feature of complex activities, and involve time consuming feature selection techniques the select the best features[49, 144]. Additionally, approaches using handcrafted features make it very difficult to compare between different algorithms due to different experimental grounds and encounter difficulty in discriminating very similar activities [149]. In recent years, deep learning techniques are increasingly become more and more attractive in human activity recognition.

Indeed, they have shown many advantages over the machine learning, among them the capability to automatically extract features. In particular, depending on the depth of the algorithm, it is possible to achieve a very high abstraction level for the features, despite machine learning techniques [138]. In these terms, deep learning techniques are considered valid algorithms to overcome machine learning dependency on the feature extraction procedure and show crucial advantages in algorithm performance.

The next paragraph is dedicated to the traditional deep learning algorithms description.

2.3.1.2 Traditional Deep Learning

Generally, the relation between input data and labels is very complex and mostly non-linear. Among Artificial Intelligence algorithms, Artificial Neural Network (ANN) is a set of supervised machine learning techniques which emulate human neural system with the aim at extracting non-linearity relations from data for classification. Human neural system is composed by neurons (about 86 billions) which are connected with synapses (around 10^{14}). Neurons receive input signals from the outer, e.g. visual or olfactory, and based on the synaptic's strength they fire and produce some output signals to be transmit to other neurons. Artificial Neural Network bases on the same neurons and synapses concept.

Figure 2.9 depicts this analogy. On the top left one human brain's neuron, on the button left a 3D simulation of a fly's brain neurons (around 25000 neurons) [142]. On the right side their mathematical models. We briefly explain the structure of the ANN depicts on the right button side.

Each data input value is associated to a neuron and its synapses strength is measured by a functional combination of input data $\mathbf{x}$ and randomly chosen *weights* $\mathbf{w}$. This value is passed to an *activation functions* σ which is responsable to determine the synapse strength and eventually to fire the neuron. The output of the activation function is given by $y = \sigma(\mathbf{w}^T\mathbf{x})$. If it fires, the output becomes the next neuron's input, see Table 2.5 for more details about activation functions.

A set of neurons is called layer. A set of layers and synapses is called network. The input data $\mathbf{x}$, is passed from the first layers to the last layer, called, respectively the *input layer* and the *output layer*, through intermediary layers, called *hidden layers*. The term Deep Learning comes from to the network's depth, i.e. when the number of hidden layers grows. Neurons belonging to same layers are not communicating to each others, while neurons belonging to different layers are connected and share the information passed through the activation function. If each neuron of the previous layer is connected to all neurons of the next layer, the former is called *fully connected* or *dense* layer.

The output layer, also called *classification layers* in case of classification task or *regression layer* in case of continuous estimation, is responsable to estimate the predicted value $\hat{\mathbf{y}}$ of the labels $\mathbf{y}$. Once the last output is computed, the *feed-forward* procedure is completed. At this point the classification error is estimated through the *loss-function* as a measure of the difference between the predicted value $\hat{\mathbf{y}}$ and the labels $\mathbf{y}$, see Table 2.6 for loss function's functional forms.

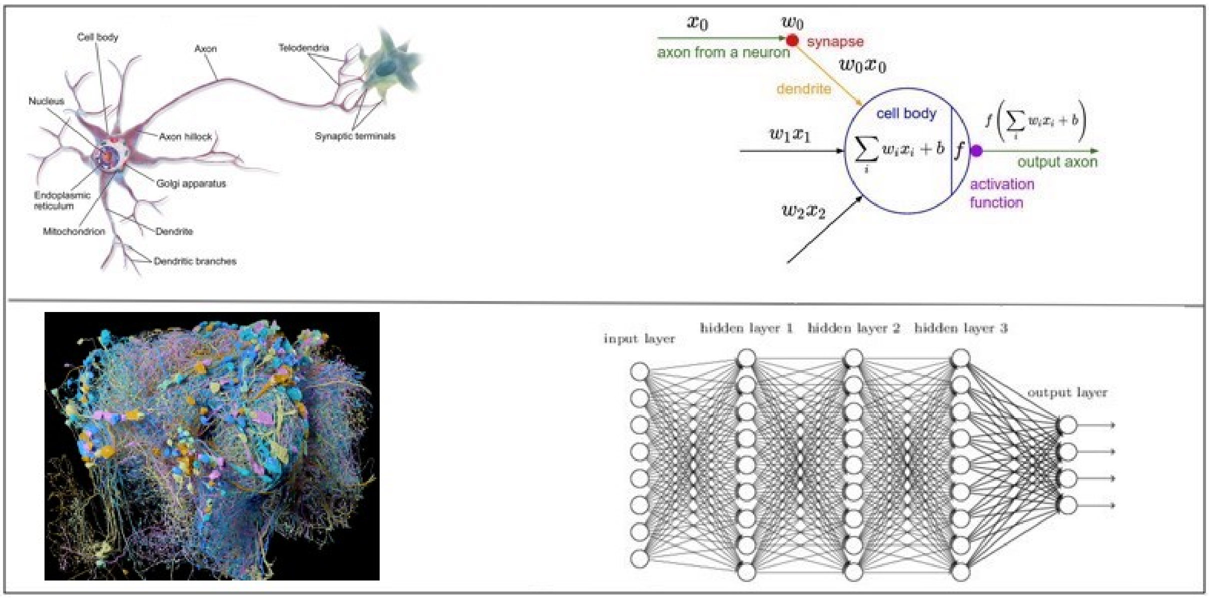


Figure 2.9: From a real neural system to Neural Networks.

Thereafter, an iterative procedure is computed to minimize the loss-function. This procedure is called *back propagation* and is responsible to minimize the loss function with respects to the weights w_i. The weight's values, indeed, represent how strong is the relation between neurons belonging to different layers and how far the input information has to be transferred through the network.

The minimization procedure bases on *gradient descent* algorithms, which iteratively search for weights, that reduce the value of the gradient of the loss until it meets the global minimum or a stopping criteria. In general, a greedy-wise tuning procedure over the hyper-parameters is performed to the aim at achieving the best network configuration. Most important hyper-parameters are: the number of layers, the kernel's number and size, the pooling's size, and the regularization parameter, such as the learning rate. As discussed in Chapter 4 we performed a grid search procedure over the layer's and the features maps number.

According to Figure 2.8, most used deep learning algorithms are described in the following:

Multi-Layer Perceptron (MLP) is the most widely used Artificial Neural Network (ANN) which are a collection or neurons organized in a layers structure, connected in an acyclic graphs. Each neuron that belongs to a layer produces an output which becomes the input of the neurons of the next adjacent layer. Most common layer type is the fully connected layer, where each neurons share their output with each adjacent layer's neuron, while same layer's neurons are not connected. MLP is made up of the input layer, one (or more) hidden layer and the output layer [145]. Used in HAR as baseline deep learning techniques, it has been often compared with machine learning, such as SVM [126, 15], RF [15], k-NN [126], DT [126], and deep learning techniques, LSTM [92], CNN [92, 126].

Convolutional Neural Networks(ConvNet or CNN) is a class of ANN based on convolution products between *kernels* and small patches of the input data of the layer. The input data is organized in channels if needed, e.g. in tri-axial accelerometer data each axes is represented by one channel, and normally convolution is performed independently on each channel. The convolutional function is computed by sliding a convolutional kernel of the size of $m \times m$ over the input of the layer. That is, the calculation of the l-th convolutional layer is given by

$$x_i^{l,j} = f\left(\sum_{a=1}^{m} w_a^j \cdot x_{i+a-1}^{l-1,j} + b_j\right)$$

where m is the kernel size, $x_i^{l,j}$ is the j-th kernel on the i-th unit of the convolutional layer l. w_a^j is the convolutional kernel matrix and b_j is the bias of the convolutional kernel. This value is mapped through the activation function σ. Thereafter, a pooling layer is responsable to compute the maximum or average value on a patch of the size $r \times r$ of the resulting activation's output. Mathematically, a local output after the max pooling or the average pooling process is given by

$$\text{max pooling} \qquad x_i^{l,j} = \max_{a,b=1}^{r}(x_{a,b})$$

$$\text{average pooling} \quad x_i^{l,j} = \frac{1}{r^2}\sum_{a,b=1}^{r}(x_{a,b})$$

The pooling layer is responsable to extracts important features and to reduces the data size's dimension.

This convolutional-activation-pooling layers block can be repeated may time if necessary. The number of repetition time determines the depth of the network. Generally, between the last block and the output layer one (or more) fully-connected layer is added to perform a fusion of the information extracted from all sensor channels [145]. After the feed-forward procedure is ended, the back propagation is performed on the convolutional weights until the convergence to the global minimum or until a stopping criterion is met. Figure 2.10 depicts CNN example in HAR, with 6 channels, corresponding to xyz-acceleration and xyz-angualr velocity data, tow convolutional-activation-max pooling layers, one fully connected layer and a soft-max layer which compute the class probability given input data.

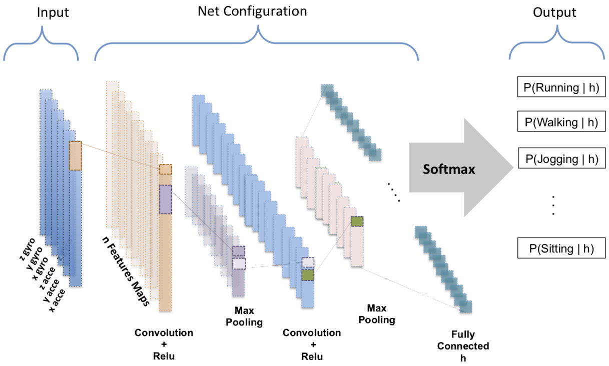


Figure 2.10: Convolutional Neural Network schema.

CNN is a very robust model under many aspects: in terms of local dependency due to the the signals correlation, in terms scale invariance for different paces or frequencies, and in terms sensor position [138, 3]. For this reasons, CNN have been largely studied in HAR. Additionally, CNN have been compared to other techniques, as follows. CNN outperforms SVM in [143] and baseline Random Forest in [76]. In [106] they demonstrate that CNN outperforms state-of-the-art techniques, which are all using hand-crafted features. More recently ensemble classification algorithm with CNN-2 and CNN-7 shows better performance when compared with machine learning random forest, boosting and traditional CNN-7 [149].

Residual Neural Networks (ResNet) is a particular convolutional neural network composed by blocks and skip connections which permit to increase the number of layers in the network. Success of Deep Neural network has been accredited to the additional layer, but authors in [59] empirically showed that there exists a maximum threshold for the network's depth without avoiding vanishing \ explosion gradient's issues.

In Residual Neural Networks the output x_{t-1} is both passed as an input to the next convolutional-activation-pooling block and directly added to the output of the block $f(x_{t-1})$. The former addiction is called *shortcut connection*. The resulting output is $x_t = f(x_{t-1}) + x_{t-1}$. This procedure is repeated many times and permit to deepen the network without adding neither extra parameters nor computation complexity. En example of ResNet is shown in Figure 2.11. Authors in [19] state that ResNet represents the most performing network in the state of the art, while authors in [40, 42] demonstrated that ResNet outperforms traditional machine learning techniques.

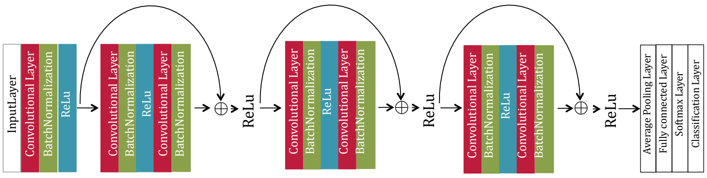


Figure 2.11: ResNet full schema.

Activation Function	*Step*	*Sigmoid*	*tanh*	*ReLU*
Formula	$\begin{cases} 0 & if\ x < 0 \\ 1 & if 0 \leq x \end{cases}$	$\frac{1}{1+e^x}$	$tan(x)$	$\max(0, x)$

Table 2.5: Activation functions.

Loss Function	*Cross-Entropy*	*Hinge*	*Euclidian*	*Absolute value*
Formula	$-\sum_{y=1}^{M} y \cdot log(p_{x,y})$	$\max(0, 1 - \hat{y} \cdot y)$	$\sum_{y=1}^{M} (\hat{y} - y)^2$	$\sum_{y=1}^{M} \lvert \hat{y} - y \rvert$

Table 2.6: Loss functions for Neural Networks.. M = number of classes, x = input data, y = class, $p_{x,y}$ = probability of being y given x.

Long-Short-Term-Memory Networks is a variant of the Recurrent Neural Network which enables to store information over time in an internal memory, overcoming gradient's vanishing issue. Given a sequence of inputs $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, LSTM's external inputs are its previous cell state c_{t-1}, the previous hidden state h_{t-1} and the current input vector x_t. LSTM associates each time step with an input gate, forget gate and output gate, denoted respectively as i_t, f_t and o_t, which all are computed by applying an activation function of the linear combination of weights, input x_i and hidden state h_{t-1}. An intermediate state $\tilde{c}_i$ is also computed through the *tahnh* of the linear combination of weights, input x_i and hidden state h_{t-1}. Finally, the cell and hidden state are updated as

$$c_t = f_t \cdot \tilde{c}_t + i_t \cdot \tilde{c}_t$$

$$h_t = o_t \cdot thanh(c_t)$$

The forget gate decides how much of the previous information is going to be forgotten. The input gate decides how to update the state vector using the information from the current input. Finally, the output gate decides what information to output at the current time step [90]. Figure 2.12 represents the network schema. Although LSTM is a very powerful techniques when data temporal dependencies have to be considered during classification, it takes into account only past information. Bidirectional-LSTM (BLSTM) offers the possibility to consider past and future information. In [57], The authors illustrate how their results based on LSTM and BLSTM, verified on a large benchmark dataset, are the state-of-the-art.

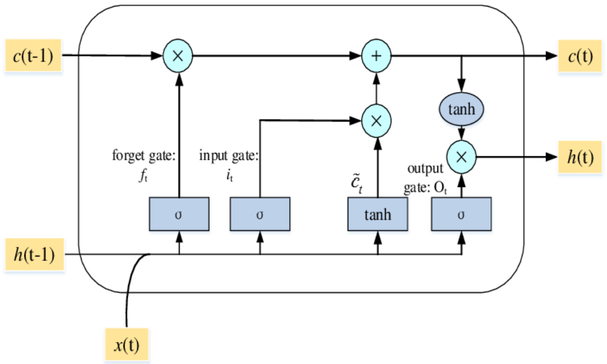


Figure 2.12: LSTM Networks schema.

During the last decade, a plenty of traditional machine learning as well as deep learning methods have been proposed for HAR [70, 95, 144, 107, 138]. Both kind of methods have been largely used in HAR and still there exists no evidence about the best methods. Traditional machine learning methods have been firstly used and have produced high performant results. Nonetheless, traditional machine learning methods present limitations related to the features extraction dependency on expert knowledge.

More recently, the advent of deep learning has widely modified the approaches in signal processing and features extraction fields [22]. First applied to 3D and 2D context in particular in vision computing domain [35, 85], deep learning methods have been shown to be valid methods also adapted to the 1D case, i.e. for time series classification [75], such as HAR. In particular, deep learning techniques enable automatic high-level feature extraction [138] and present many advantages over the traditional machine learning techniques, as mentioned in the above sections. In the most recent literature, deep learning methods have been become predominant [138].

However, deep learning techniques, unlike traditional machine learning approaches, require a large number of samples and an expensive hardware to estimate the model [42]. Large scale inertial datasets with millions of signals recorded by hundreds of human subjects are still not available, and instead several smaller datasets made of thousands of signals and dozens of human subjects are publicly available [117]. It is therefore not obvious in this domain, which method between deep and machine learning is the most appropriate, especially in those case where the hardware is low cost.

Scarcity of data results in an important limit of machine learning and deep learning approaches in activities classification: the difficulties in being able to generalize the models against the variety of movements performed by different subjects [18]. This variety occurs in relation to heterogeneity in the hardware on which the inertial data is collected, different inherent capabilities or attributes relating to the users themselves, and differences in the environment in which the data is collected. One of the most relevant difficulty to face with new situations is due to the *population diversity* problem [72], that is, the natural differences between users' activity patterns, which implies that different executions of the same activity are different. A solution is to leverage labeled user-specific data for a personalized approach to activity recognition [26]. Personalization methods are presented in the next Section 2.3.2.

2.3.2 Personalized Learning Methods

Although research on activity recognition techniques from wearable devices is very active, the traditional systems are limited in their ability to generalize to new users and/or new environments, and require considerable effort and customization to achieve good performance in a real-context [61, 63].

One of the most relevant difficulty to face with new situations is due to the *population diversity* problem [72], that is, the natural differences between users' activity patterns, which implies that different executions of the same activity are different.

Ideally, algorithms should be trained on a representative number of subjects and on as many cases as possible. The number of subjects present in the data set does not just impact the quality and robustness of the induced model, but also the ability to evaluate the consistency of results across subjects [83]. Furthermore, although new technology potentially enable to store large amount of data from varied devices, the real availability data is very scarce and public datasets are normally very limited, see Section 2.2. In particular, it is very difficult to source labeled data necessary to train supervised machine

learning algorithms. To face this problem, activity classification models should be able to generalize as much as possible with respect to the final user.
In the following sections we discuss state-of-the-art results related to *population diversity* issue based on the personalization of machine learning and deep learning algorithms.

2.3.2.1 Personalized Machine Learning

In order to achieve generalizable activity recognition models based on machine learning algorithms, three approaches are mainly adopted in literature:

- **Data based approach** encompass three data split configurations: *subject - independent*, *subject - dependent*, and *hybrid.*The *subject - independent* (also called *impersonal*) model does not use the end user data for the development of the activity recognition model. It is based on the definition of a single activity recognition model that must be flexible enough to be able to generalize the diversity between users and it should be able to have good performance once a new user is to be classified. The *subject - dependent* (also called *personal*) model only uses the end user data for the development of the activity recognition model. The specific model, being built with the data of the final user, is able to capture her/his peculiarities, thus it should well generalize in the real context. The flaw is that it must be implemented for each end user [16]. The *hybrid* model uses the end user data and the data of the other users for the development of the activity recognition model.

  In other words the classification model is trained both on the data of the users and on a part of the data of the final user. The idea is that the classifier should recognize easier the activity performed by the final user. Figure 2.13 shows a graphical depiction of the three models to better clarify their differences. Tapia et al. [130] introduced the subject - independent and subject - dependent models, and later Weiss at al. [140] the hybrid model.
  The models were compared by different researchers and also extended in order to achieve better performance. Medrano et al. [87] demonstrated that the subject - dependent approach achieves higher performance then subject - independent approach for falls detection, called respectively *personal* and *generic fall detector*.
  Shen et al. [110] achieved similar results for activity recognition and come to the conclusion that the subject - dependent (termed *personalized*) model tends to perform better than the subject - independent (termed *generalized*) one because user training data carries her/his personalized activity information.
  Lara et al. [74] consider subject - independent approach more challenging because in practice, a real-time activity recognition system should be able to fit any individual and they consider not convenient in many cases to train the activity model for each subject.
  Weiss at al. [140] and Lockhart et al. [82] compared the subject - independent and

the subject - dependent (termed *impersonal* and *personal* respectively) with the hybrid model. They concluded that the models built on the subject - dependent and the hybrid approaches achieve same performance and outperform the performance of the model based on the subject - independent approach.
Similar conclusions are achieved by Lane *et al.* [72], who compare subject - dependent and subject - independent (respectively named *isolated* and *single*) models with another model called *multi-naive.* In this case, subject - dependent approach outperformed the other two approaches as the amount of the available data increases.
Chen et al. [31] compared the subject - independent, subject - dependent, and hybrid (respectively called *rest-to-one*, *one-to-one*, and *all-to-one*) models, and once again the subject - dependent model outperforms the subject - independent model, whereas the hybrid model achieves the best performance. The authors also classify subject - independent and hybrid models as *generalized* models, while the subject - dependent model falls into the category of the *personalized* models.
Same results have been achieved by Vaizman et al. [132], who compared the subject - independent, subject - dependent, and hybrid (respectively called *universal*, *individual*, and *adapted*) models. Furthermore, they introduced context-based information by exploiting many sensors, such as, location, audio, and phone-state sensors.

- **Similarity based approach** which consider the similarity between users as crucial factor for obtaining a classification model able to adapt to new situations. In these direction, Sztyler et al. [128, 129] proposed a personalized variant of the hybrid model. The classification model is trained using the data of those users that are similar to the final user based on signal patterns similarity. They found that people with same fitness level also have similar acceleration patterns regarding the running activity, whereas gender and physique could characterize the walking activity.
  The heterogeneity of the data is not eliminated but it is managed in the classification procedure.
  A similar approach is presented by Lane et al. [72]. The proposed approach consists in exploiting the similarity between users in order to weight the collected data. The similarities are calculated based on signal pattern data, or on physical data (e.g., age and height), or on lifestyle index. The value of similarity is used as weight. The higher the weight, the more similar two users are and the more that signals from those users is used for classification.
  Garcia-Ceja et al. [50, 51] exploited inter-class similarity instead of the similarity between subjects (called inter-user similarity) presented by Lane et al. [72]. The final model is trained using only the instances that are similar to the target user for each class.

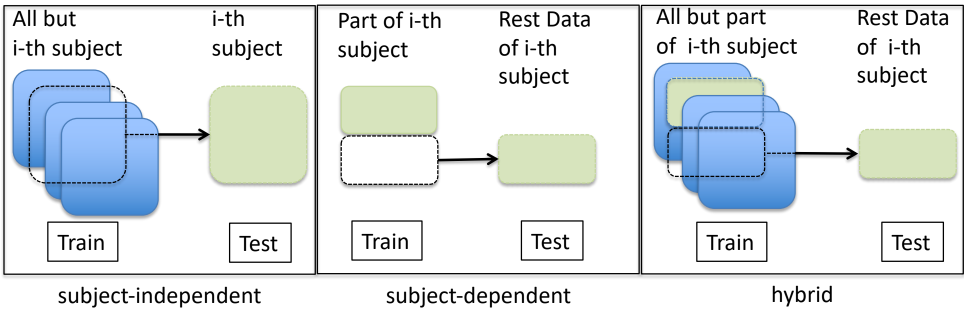


Figure 2.13: A graphical representation of subject-independent, subject-dependent and hybrid models.

- **Classifier based approach** obtains generalization from several combinations of activity recognition models. Hong at al. [61] proposed a different solution where the generalization is obtained by a combination of activity recognition models (trained by a subject - dependent approach). This combination permits to achieve better activity recognition performance for the final user. Reiss et al. [102] proposed a model that consists of a set of weighted classifiers (experts). Initially all the weights have the same values. The classifiers are adapted to a new user by considering a new set of suitable weights that better fit the labeled data of the new user.

2.3.2.2 Personalized Deep Learning

Personalized Deep learning for heterogeneity with users in activity recognition have been explored in the literature and mainly refer to two approaches:

- **Incremental Learning** refers to recognition methods that can learn from streaming data and adapt to new moving style of a new unseen person without retraining [121]. Yu et al. [146] exploited the hybrid model and compare it to a new model called *incremental hybrid model*. The latter is trained first with the subject - independent approach and then it is incrementally updated based on personal data from a specific user. The difference from the hybrid is that the incremental hybrid model gives more weights to personal data during training.
  Similarly, Siirtola et al. [119] proposed an incremental learning method. The method initially uses a subject - independent model, which is updated with a 2-steps feature extraction method from the test subject data. Afterwards, the same authors proposed a 4 steps subject - dependent model [118]. The proposed method initially uses a subject - independent model, collects and labels the data from the user based on the subject - independent model, trains a subject - dependent model on the collected and labelled data, and classifies activity based on the subject - dependent

model.
Vo et al. [136] exploited a similar approach. The proposed approach first trains a subject - dependent model from data of subject A. The model of subject A is then transferred to subject B. Then, the unlabelled samples of subject B are classified to the model of subject A. These data are finally used to adjust model for the subject B.
Abdallah et al. [1] propose an incremental learning algorithm based on clustering procedure which aims at tuning the general model to recognize a given user's personalised activities.

- **Transfer Learning** bases on pre-trained network, it adjusts weights using new user's data. This procedure permits to reduce the time consumption of the training phase. In addition, it is a powerful method for when scarcity of labeled data does not permit to train a network from scratch. Rokni et al. [104] propose to personalize their HAR models with transfer learning. In the training phase, a CNN is firstly trained with data collected from a few participants. In the test phase, only the top layers of the CNN are fine-tuned with a small amount of data for the target users.

Personalized machine learning has been largely investigated in literature and many different approaches have been proposed. In particular, personalized machine learning methods emphasize the user's perspective, e.g. the models are modified in order to involve user's physical, sensors characteristics and her\him intra-, inter-variability. In contrast, state-of-the-art deep learning approaches do not concentrate on subject variability by explicitly computing extra evaluation on the user's characteristics, and user's heterogeneity is left to the weights update of the network. In other words, deep-learning methods focus rather on update and slightly modify pre-trained models when new user's data are available. The capability of deep-learning techniques to extract very high level features from data overcomes the necessity of additional information and intrinsically user's variability is extracted directly from data.

2.3.2.3 Evaluation of the Classification performance

The evaluation of the classification performances aims at evaluate the reliability of the results. For now on, we focus on supervised machine learning methods because is the core of this thesis.

In supervised machine learning algorithms, the classification based on the definition three datasets: the training, the validation and the test datasets. The training set is designed to estimate the relation between input and output, together with the model parameters. The validation set is designed to affine and tune the model parameters and hyper-parameters. With hyper-parameters, it is meant the parameters which are not necessarily directly involved in the model, but define the structure of the algorithm, such as, for instance, the number of the channels in a deep network. Finally, the test set is used to evaluate the classification performance of the resulted classification model.

Training, validation and test sets are generally defined as a partition of the original dataset and mostly representing, for instance, the 70%, 20%, and 10% of the number of the samples. It is a common practice to perform the *k-fold cross-validation* procedure [114, 15, 105]. The k-fold-cross-validation is a procedure that helps to achieve more robust results and helps to avoid that the algorithm specializes on a specific partitions of the original dataset.

In particular, it consists in split the training and test set in k-folds. The entire classification procedure is performed on each split, k times. Thus, k models are estimated, and their performances are evaluated and averaged. Especially in HAR, there are several variants of k-fold cross-validation. More in details, HAR community has defined three main approaches: *subject-independent, subject-dependent*, and *hybrid* [130] [140] [44]. The *subject-independent* (also called impersonal) approach does not use the end user data for the development of the activity recognition model. That is, the training set contains all subject but the end-user subject. The *subject-dependent* (also called personal) approach only uses the end user data for the development of the activity recognition model. Thus, the training and test set collect only data of end-user.

The flaw of this approach is that it must be implemented for each end user . The *hybrid* approach uses the end user data and the data of the other users in the training and test set. The classification model is trained both on the data of the users and on a part of the data of the final user. The idea is that the classifier should recognize easier the activity performed by the final user.

The classification performance is calculated through heuristic metrics based on the correctly classified samples. In particular, these metrics are all derived from the *confusion matrix*. In supervised machine learning, the *confusion matrix* compares the groundtruth (the observed labels) with the estimated labels. The binary case is shown in Table 2.7

	Estimated	
Groundtruth	1	0
1	True Positives (TP)	False Negatives (FP)
0	False Positives (FN)	True Negatives (TN)

Table 2.7: The Confusion Matrix: a representation of true negative, true positive, false negative and false positive.

True Positives are observed 1-class samples which are classified as 1. True Negatives represent the number of observed 0-class samples which are classified as 0. False Negatives are 0-class samples which are classified as class 1. Viceversa, False Positives represent the

number of samples classified as 1-class but which truly belongs to 0-class. The confusion matrix can be extended to the multi-class classification problem. In this case, on the principal diagonal are displayed the number of correctly classified samples, while out of the principal diagonal miss-classified samples are listed.
The classification performance can be measured by focusing either on the number of correct classified samples or by giving more importance on the miss-classification. The choice of the evaluation metric accentuates either one or the other aspect of the classification. In the context of HAR, the *accuracy* is the most used metric for the evaluation of the classification performance [72, 124, 60, 149]. According to the confusion matrix showed in Table2.7, accuracy (Acc) is defined as follows:

$$Acc = \frac{TP + TN}{TN + FP + FN + TP}$$

It calculates the percentage of correctly classified samples over the total number of the samples. The accuracy highlights the correct classification performance and gives more emphasis to the classification of the true positives and of the true negatives.

In some cases, it is required that the evaluation of the classification performance accentuates the miss-classifications, such as either the false positive or the false negatives cases. For instance, in the case of falls detection, the algorithm should be more penalized if it does not recognize a fall, when it occurs (false negative) instead that it does recognize a normal behavior as fall (false positive). An appropriate metric for this case is the *F_β-score*. It is defined as function of the *recall* and *precision*. The *recall* is also called sensitivity or true positive rate and is calculated as the number of correct positive predictions divided by the total number of positives, the best value correspond to 1, the worst to 0. The *precision* is also called positive predictive value and is calculated as the number of correct positive predictions divided by the total number of positive prediction, the best precision is 1 whereas the worst is 0. Formula are given by:

$precision = \frac{TP}{TP+FP}$ percentage of true positive among the positive classified samples

$recall = \frac{TP}{TP+FN}$ percentage of true positive among the real number of positive samples

$F_\beta = \cdot \frac{(1+\beta^2) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall}$ combination of precision and recall measures

If $\beta = 1$, F_1-score is the harmonic mean of the precision and the recall.

The *specificity*, also called true negative rate (TNR), is calculated as the number of correct negative predictions divided by the total number of negatives. Best value corresponds to 1, while the worst is 0. Together with the sensitivity, the specificity helps to determine best parameter value, when a tuning procedure is computed. A common practice is to compute the Receiver Operating Characteristic curve (ROC) which plots the value of the sensitivity against the 1-specificity by varying the values of the model parameters. The curve is valid instrument which takes into account both the sensitivity and the specificity for the models parameter choices. From the ROC it is possible to compute the Area Under the Curve (AUC), which represents another measure of the classification performance. In particular, it provides an aggregated measure of the algorithm performance across all possible classification models. It can also be interpreted as the probability that the model ranks a random positive example more highly than a random negative example.

2.3.3 Conclusions

Today portable devices have advanced computing capability and connectivity and usually include several sensors, such as a accelerometer and gyroscope, which provide a considerable amount of data. Those factors have stimulated the interest of the scientific community in developing artificial intelligence methods for automatic Human Activity Recognition. In particular, HAR community focuses on machine learning and deep learning techniques. In Section 2.3.1, state-of-the-art traditional methods have been presented.

As discussed, traditional methods presents several limitations. Traditional machine learning methods are low cost in terms of time consumption, data, and complexity, but the dependency on expert knowledge in the features extraction phase generates non-robust models, which are often difficult to compare. Deep learning methods remains stable in terms of feature extraction, which is mainly automatically done, but the training phase requires much more data, and, consequently, it is either very time consuming or requires expansive hardware. Particularly in HAR, traditional machine learning and deep learning systems are limited in their ability to generalize to new users and/or new environments, and require considerable effort and customization to achieve good performance in a real-context due together to the inter-\ intra-subject variability and to the scarcity of data.

Personalized machine learning and deep learning state-of-the-art solutions have been reported. In particular, state-of-the-art machine learning solutions mostly refer to three personalization concepts: data-based personalization, similarity-based personalization and classifier-based personalization. That is, a preprocess is made before training the model, aimed at managing subject's variability, based either on different training and test datasets split or on user's similarity metrics. The preprocess, generally, improves the algorithms capability to discriminate between subjects and their differences. A new instance becomes easier to classify because of additional user's information which address the algorithm choices.

In the following chapters we will discuss in more details about machine learning and deep learning techniques and their personalization. In particular, the work is subdivided in two main chapters. A new personalized machine learning method is defined in chapter 3 and compared to traditional machine learning methods.

In Chapter 4 we evaluate deep learning methods for HAR and propose a new personalized deep learning methods based on subjects similarity and compare the results with the traditional and personalized machine learning techniques.

Chapter 3

Personalization in HAR and Machine Learning

3.1 Motivation

During the last decade, plenty of traditional machine learning as well as deep learning methods for HAR that use accelerometers have been proposed in literature [112, 73, 144, 107, 138]. However, real-world HAR systems achieve non satisfying recognition accuracy in real world applications mostly because HAR techniques struggle to generalize to new users and/or new environments [61, 63].

Several factors may affect the accuracy of activity recognition methods: i) position of the device: pocket, hand, bag, etc; ii) differences between different brands of sensors, in terms of sensitivity range and sampling frequency; iii) human characteristics, such as age, gender, weight, height, lifestyle, and physical abilities. While factors related to the position and the characteristics of the devices have been largely investigated, few works have explored the effects of human characteristics on recognition accuracy [10, 72, 140, 87, 61].

Lane et al. [72] proposed a new method to take into account human factors. This method exploits the similarity between users to weight training data and thus to improve the recognition accuracy. Unfortunately, results achieved by these researchers are not reproducible because the dataset used for the experimentation is not publicly available and moreover, the authors mainly focused on the automatic annotation of inertial signals and not classification of activities of subjects. The approach proposed by Lane et al.[72] deserves further investigation and thus it has been the starting point of the research we performed and whose results are presented in this section. Our research question was: *does personalization machine learning methods be employed for increasing the accuracy of traditional machine learning techniques based on accelerometer signals?* To reply to this research question, we have:

- experimented several personalization methods on three public datasets, MobiAct, Motion Sense and UniMiB-SHAR, in order to make the results reproducible and thus allowing future research on this topic;

- defined a suitable procedure to split the experimental data into training, validation, and test sets;

- defined a new personalization method that includes and generalizes the different ideas discussed in the state-of-the-art.

The Chapter is organized as follows. Section 3.2 presents the proposed models. In particular, section 3.2 describes the Activity Recognition Process (ARP) applied on three dataset, the implementation of the personalization models, and the classification algorithms we exploited for the analysis. Section 3.3 describes the experiment setup in which the description of the datasets is presented together with a preliminary statistical analysis. Section 3.4 discusses a comparison between personalized machine learning and traditional machine learning techniques. Finally, section 3.5 the conclusions.

3.2 Proposed Methods

In this section we propose personalization models based on similarity between users in term of physical attributes and/or signals patterns. Personalization models are used to weight users training data of the classifier, that in our case is the Adaboost classifier. We demonstrate that a classifier trained on data personalized in this way is more powerful, in terms of recognition accuracy, with respect to a classifier trained without personalization.

The rationale behind similarity-based personalizations is based on two intuitions:

1. Users with different physical characteristics, such as age or weight, walk or run in a different way. This results in a different accelerometer signal. Focusing the training data on those users that are more similar to the user under test may help to increase recognition capabilities of the classifier. We refer to this aspect as *physical-based similarity.*

2. Independently from similarities based on physical characteristics, accelerometer signals from two different users may be more similar with respect to other users performing the same activity. We refer to this aspect as *signal-based (or sensor-based) similarity.*

Before entering into the core of the methods we propose, we introduce a method of data split for training and testing data that we consider an indispensable step in order to reliably validate the methods of personalization under test. Figure 3.1 shows the steps of the method that are described in the following.

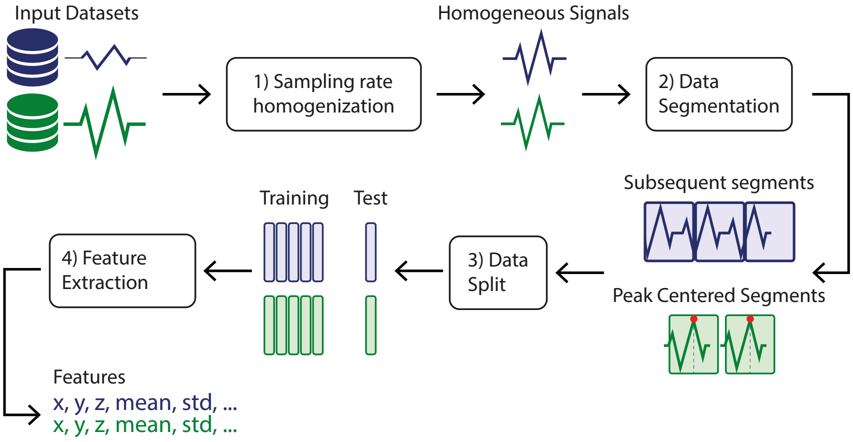


Figure 3.1: Data preparation and feature extraction pipeline.

3.2.1 Data preparation

For validating personalization models we selected three different datasets: UniMiB-SHAR [88], MobiAct [135], and Motion Sense [86]. Each signal of the datasets is composed of three accelerometer components along the x, y, and z axis. Since public available datasets are usually acquired by different research groups using different devices and protocols, it is common to have non-homogeneous characteristics in terms of data collection, such as position of the sensor, sampling frequency of the sensor, number of participants, activities not performed by all users, etc. This inhomogeneity leads sometimes to not-comparable results among different datasets.

Machine learning methods take a segment made of N subsequent accelerometer samples as input. It means that the original accelerometer signals need to be segmented before being fed into a classifier.

Another important point concerning data preparation is represented by the split between training and test set. In literature there are two very common procedures, that are the k-cross and leave-one-subject-out validation. Those are mutually exclusive. As described later on, for our scope, we had to employ the k-cross validation on the top of the leave-one-subject-out one.

Defining a common protocol for data preparation is necessary in order to manage different datasets and it is important for the reproducibility of the experiments.

In the following we describe the steps of the data preparation protocol that are pictured in Figure 3.1.

3.2.1.1 Sampling rate homogenization

Sensors acquire data at a given sampling frequency, with a given range of intensity values, and so on. Each manufacturer designs its own sensor with operating specifications that may be different from the typical ones. Moreover, sensors acquire data at not constant sampling frequency. The three datasets we considered have different sampling rates. Since machine learning methods require input data in a given format (e.g., number of samples per second and intensity range), that is, consistent over time, we had to pre-process data from the datasets in order to make them homogeneous in terms of sampling rate. To this end, we have chosen the lowest sampling rate among all the sampling rates of the datasets, then we have subsampled all the signal to fit that frequency. The sampling rate chosen is 50 Hz. Literature suggests that about 50Hz is a suitable sampling rate that permits to model human activities [101], thus our choice does not negatively affect the results. This step is represented by the action number 1 in Figure 3.1.

3.2.1.2 Data segmentation

It is a common practice to divide the original signal data in segments, or windows, which contain a certain number of samples taken from the original accelerometer signal at a given frequency rate. The length of each segment is a parameter of the classifier and must be compatible with the temporal duration of the activities. In other words, the segment should contain at least one occurrence of any activities included in the dataset list. Otherwise, signals that contain incomplete portions of activity may be erroneously classified. Usually, the slowest activity is walking and it is common practice to consider 2.5 seconds as the minimum temporal interval to observe two human steps. It means that, a 2.5 seconds segment, at a sampling frequency of 50 Hz, contains 125 samples.

Once the length of each segment has been fixed, there are two possible ways for segmenting data:

- *Subsequent (overlapped) segments.* Each signal is divided into subsequent windows of a given size and each of them can be overlapped with the previous and the next one. The overlapping percentage is a parameter of the algorithm. In this study we considered windows of 5 seconds. We computed the analysis with no overlapping and with 50% of overlapping and compare the results [88].
- *Segments centered on the peak of the signal.* Each signal is divided into subsequent windows but only if the intensity of the recorded signal is higher than a given threshold. Usually the threshold is $2g$, where g is the gravitational constant. When a value v exceeds the threshold, a segment is taken around the temporal position of the peak value v.

Data segmentation allows to augment data and consequently to reduce the overfitting of machine learning algorithms that is most of the time caused by a low amount of training data. In Figure 3.1 we show the data segmentation (action number 2) with a dataset with subsequent segments (the blue dataset) and another dataset that exploits peak centered segments (the green dataset).

3.2.1.3 Data split

A human activity dataset is composed of inertial signals recorded through a smartphone worn by human subjects that during experimental sessions performed a certain number of activities. Ideally, there is a list of activities to be performed by all the subjects. One of the recurring problems with the literature datasets is that not all the subjects performed all the activities that are in the dataset list.

The k-fold cross validation separates the dataset into k groups which, alternately, compose the training and the test sets disregarding if the segments of a given subject is either in the training and in the test split (shown in Figure 3.1 as action number 3).This behavior is not suitable for investigating personalization models, because we have to be sure that data from the user under test are or not within the training set. To this end, the subject-out-validation strategy is a more suitable way to split training and test data. This strategy considers training data made of segments from all the subjects but the subject under test.

To better explore personalization methods, we also need to ensure that the training and test splits are composed by the same list of activities and more important that all the users of dataset performed the same list of activities. This is not always true, because, as discussed above, it may happen that not all the subjects performed all the activities of the list. To this scope, we removed those subjects from the dataset that did not perform all the activities included in the dataset list or alternatively, if this leads to a huge lost of subjects, we removed from the dataset those activities not performed by all subjects.

In this study we explored personalization methods on the top of three different data splits: *subject-dependent*, *subject-independent* and *hybrid* (see Figure 2.13). The subject-dependent split considers the training and test sets made of only signals from the subject under test. The subject-independent split considers a test set made of only signals from the subject i, and a training set made of signals from all the subjects but i. The hybrid split is a subject-independent split with the injection of a small amount of signals of the subject i within the training set. In order to realize the hybrid approach we need that some data from the subject under test is part of the training data. At the same time, to make the hybrid approach comparable with the others, we need the test set is always the same whatever is the data-split adopted. To ensure that, the data from each subject is divided using the k-fold cross validation strategy with the constraint that each fold contains the same number of activities. Given a subject under test, one of the k folds is taken as test set for all the three data-splits: *subject-dependent*, *subject-independent* and *hybrid*. The remaining folds are used as training data of the *subject-dependent* data split, and also in combination with the data from other subjects in the *hybrid* data split. The training data of the *subject-independent* is made of all the $k-1$ folds of each subjects that are not included in the corresponding test sets.

3.2.1.4 Feature extraction

In literature it is shown that using a set of features instead of raw data improves the classification accuracy [40]. Furthermore, features extraction reduces the data dimensionality while extracting the most important peculiarity of the signal. Each accelerometer signal of the datasets is composed of three accelerometer components along the x, y, and z axis. An entire segment is as follows:

$$\big(\underbrace{acc_{x_1},\ acc_{x_2}\ acc_{x_3}\ \dots\ acc_{x_n}}_{x-dimension\ acceleration}\cdots\underbrace{acc_{y_1}\ acc_{y_2}\ acc_{y_3}\ \dots\ acc_{y_n}}_{y-dimension\ acceleration}\cdots\underbrace{acc_{z_1}\ acc_{z_2}\ acc_{z_3}\ \dots\ acc_{z_n}}_{z-dimension\ acceleration}\big) \tag{3.1}$$

where $n = s \cdot f$, and f is the sampling rate.

We considered a vector of hand-crafted time and frequency domain features calculated on each segment and for each accelerometer direction. Table 2.2 presents the features we considered [65, 84, 133, 40, 20]. The resulting feature vector is obtained by concatenating the feature extracted from the component x, y, and z of the accelerometer signal. This is presented in Figure 3.1 in action number 4 as the last step of the data preparation protocol.

3.2.2 Subject similarity

In this study we define a new personalization model including and generalizing the different models presented in the-state-of-the-art. The general idea is that people with different physical aspects, life style, or habits may walk, run, etc., in a different way and that accelerometer signals related to the same activity, disregarding the physical similarities between subjects, may have common characteristics [72].

To take into account such a diversity, we introduce the concept of similarity between subjects and than we exploit the similarity to weight the training data in order to give more importance to data that are more similar to data of the user under test.

Each subject i can be described with a feature vector $\mathbf{g}_i = \{g_1, \dots, g_K\}$. Similarity between two subjects i and j is defined as follows:

$$\mathrm{sim}(i,j) = e^{-\gamma d(i,j)} \tag{3.2}$$

where γ is a scale parameter and $d(i,j)$ is the Euclidean distance between the feature vector of two subjects:

$$d(i,j) = \sqrt{\sum_{k=1}^{K}(g_{k,i} - g_{k,j})^2} \tag{3.3}$$

The resulting similarity value ranges from 0 to 1: 0 means that the two subjects are dissimilar, and 1 means that the two subjects are equal. The idea is to take advantage of the similarity between subjects as follows. Given a subject i under test, all the training data are weighted by using the similarity between the user i and the rest of the users. We can define three types of similarity: *physical-based* ($\text{sim}^{physical}$), *sensor-based* (sim^{sensor}), and *physical combined with sensor-based similarity* ($\text{sim}^{physical+sensor}$).

Figure 3.2 shows the training data which are weighted according to the similarity. Each sample are so considered with different importance in the classification with respect to the similarity between the subject in the training and in the test set.

3.2.2.1 Similarity based on physical characteristics

For each subject we define a feature vector that is made of three real values $\mathbf{g}^{physical} = (age, weight, height) = (g_1^p, g_2^p, g_3^p)$. Each component of the triplet ranges from 0 to 1 because all the ages, weights, and heights of the subjects have been normalized to fit the range of real number $[0-1]$. The choice of these characteristics is inspired by the literature and it is subject to the availability of the metadata of the public datasets. We decided to not consider lifestyle of the subjects as further subject characteristic because this information is usually not available in public datasets.

Figure 3.2 shows some examples of physical similarity between subjects from the dataset Motion Sense. The examples have been obtained with different values of γ ranging from a low to a high value. Each figure is the visual representation of the similarity matrix between all the subjects of the dataset. Clearly the diagonal of the matrix is always 1 (each subject is similar to him/herself). The parameter γ plays a crucial role in the definition of personalization models. γ determines the shape of the exponential function: higher values of γ correspond to more separation between subjects. With $\gamma = 0$ all the subjects has similarity 1.

3.2.2.2 Similarity based on signal distance

For each subject we define a feature vector that is made of 18 real values described in Table 2.2: $\mathbf{g}^{sensor} = (g_1^s, \dots, g_{18}^s)$. Each subject i has N_i segments. We calculate the similarity between 2 subjects i and j by summing up the similarity between each segment of the subjects:

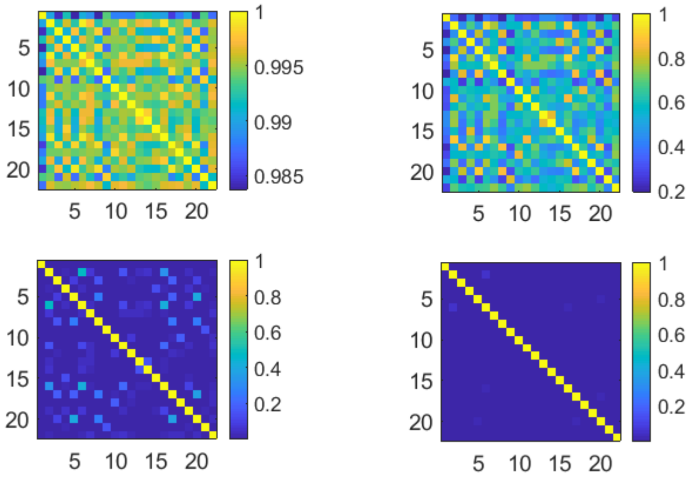


Figure 3.2: Physical Similarity Matrix for different values of $\gamma = 0.01,\ 1,\ 10,\ 40$ (clockwise order).

$$\begin{aligned}\mathrm{sim}^{sensor}(i,j) &= \sum_{m=1}^{N_i}\sum_{n=1}^{N_j} \mathrm{sim}(x_{in}, x_{jm}) \\ &= \sum_{m=1}^{N_i}\sum_{n=1}^{N_j} e^{-\gamma d(x_{in},x_{jm})} \end{aligned} \tag{3.4}$$

3.2.2.3 Similarity based on physical characteristics and signal distance

For each subject we define a similarity with respect to the other subjects that is made of the weighted sum of physical and sensor similarity:

$$\mathrm{sim}^{physical+sensor}(i,j) = \alpha \cdot \mathrm{sim}^{sensor}(i,j) + \beta \cdot \mathrm{sim}^{physical}(i,j) \tag{3.5}$$

$$\tag{3.6}$$

where α and β are such that $\alpha + \beta = 1$.

3.2.3 Personalization models

In order to evaluate the influence of the similarity-based personalization strategies, we have performed 2 groups of experiments:

- *Experiments without similarity-based personalization*: This group of experiments ignores the similarity between subjects that is equal to consider for all the types of similarities $\gamma = 0$. For this first group we considered the following dataset splits:
  - *Subject-dependent model*: the training set and the test set are composed only by the data of the test subject. For each subject under test, the procedure is reapeated k times according to the k-cross-validation.
  - *Subject-independent model*: we train the classifier using the data of all subjects but the test. For each subject under test, the procedure is reapeated k times according to the k-cross-validation.
  - *Hybrid model*: we train the classifier using the data of all subjects and a portion of the data of the test which are the $(k - 1)$ splits not used for the test. For each subject under test, the procedure is reapeated k times.
- *Experiments with similarity-based personalization*: This group of experiments considers the similarity between users by integrating the sample data with the similarity weights and so the classification is influenced by the similarity between users. For the second group of experiments, we considered the following dataset splits:
  - *Subject-independent-weighted model*: we train the classifier using the data of all subjects but the test as in the first group of experiments. The segments of the training data are weighted with the similarity between the subjects in the training and the subject of the test. For each subject under test, the procedure is reapeated k times according to the k-cross-validation.
  - *Hybrid weighted model*: we train the classifier using the data of all subjects and a portion of the data of the test (which are the $k-1$ splits not used for the test). The segments of the training data are weighted with the similarity between the subjects in the training and the subject of the test. For each subject under test, the procedure is reapeated k times according to the k-cross-validation.

  For this group of experiments, it is not possible to employ the subject-dependent split because by definition it does not contain samples from other users and then it is not possible to compute the similarity between different subjects.

3.2.4 Classifier

To evaluate the goodness of the personalization strategies we considered the Adaboost classifier which permits to weight training data before starting the training process [21]. We have also experimented Support Vector Machines and k-Nearest Neighbor. However for these classifiers the adoption of the similarity-based weighting procedure did not lead to remarkable accuracy modifications whatever was the value of γ.

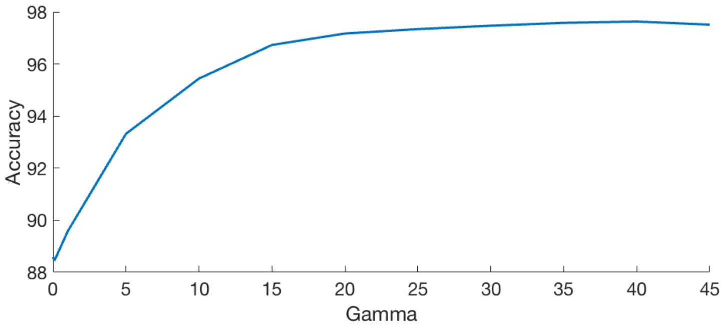


Figure 3.3: Accuracy as the values of γ increases.

The classification was reapeated several times: for each model and for each value of γ. The choice of appropriate values of γ is not trivial. The value of γ is arbitrary in (0,+∞), zero and infinity excluded.

Whatever is the classifier, the performance are measured using accuracy. Given E the set of all the activities types, $a \in E$, NP_a the number of times a occurs in the dataset, and TP_a the number of times the activity a is recognized, accuracy is define as in Equation (3.7):

$$Acc = \frac{1}{|E|}\sum_{a=1}^{|E|} Acc_a = \frac{1}{|E|}\sum_{a=1}^{|E|}\frac{TP_a}{NP_a}. \tag{3.7}$$

Acc is the arithmetic average of the accuracy Acc_a of each activity, defined in Section 2.2.

A grid searching procedure permitted us to select the best value according with accuracy results. In Figure 3.3 we observe the accuracy behavior as γ increases. We notice that if the value of γ grows, also the value of the accuracy grows until the value 40, and then starts to decrease. This behavior is exactly what we expected because when $\gamma \to +\infty$ the number of training data decreases and so the accuracy. If $\gamma = 0$ we also have a decreasing of accuracy because it correspond to a unpersonalized model.

3.3 Experiment setup

We experimented three public datasets containing accelerometer signals of Activities of Daily Living (ADLs) and Falls recorded by the sensors embedded in smartphones.

UniMiB-SHAR The dataset contains tri-axial acceleration data organized in 3s windows around the peak. Signals of 17 different activities (ADLs and Falls) are collected and

	Sex		**Age**	**Weight**	**Height**
Dataset	Male	Female	(years)	(kg)	(cm)
UniMiB-SHAR	6	22	18-60	50-82	160-190
			27 ± 11	$64.4 \pm 9,7$	169 ± 7
MobiAct	43	14	20-47	50-120	158-193
			25.19 ± 4.45	76.8 ± 14.16	175.73 ± 7.77
Motion Sense	13	9	18-46	48-102	161-190
			28.8 ± 5.46	72.12 ± 16.21	174.2 ± 8.9

Table 3.1: Datasets statistics concerning subjects physical characteristics.

performed by 30 subjects. For each of them sex, age, weight, and height are known [88]. The original sampling rate is 50Hz. We have chosen segments of 3 seconds for this dataset.

The subjects placed the smartphone used for the acquisition (a Samsung Galaxy Nexus I9250) half of the times in the left trouser's pocket and the remaining times in the right one. They repeat each activity several times. After having applied the data split procedure described in section 3.2.1.3, the number of subjects is 27 and the number of the activity is 13.

MobiAct.This dataset includes tri-axial acceleration data of 15 ADLs and Falls recorded with a Samsung Galaxy S3 and performed by 67 participants. The windows size we considered is of 5s with a sample rate of 87Hz. Additional information on the subjects are sex, age, weight, and height. The smartphone is located with random orientation in a loose pocket chosen by the subject [135]. The original sampling rate is 87Hz. We have chosen segments of 5 seconds for this dataset.

After having applied the data split procedure described in section 3.2.1.3, we removed 10 subjects and 2 activities because of missing values.

Motion Sense.This dataset contains time-series data generated by the accelerometer sensors of an iPhone 6s worn by 24 participants. Each of the subjects performed 6 activities (only ADLs). The smartphone were kept in the participant's front pocket. After having applied the data split procedure described in section 3.2.1.3, we removed 2 subjects [86]. The original sampling rate is 50Hz. We have chosen segments of 5 seconds for this dataset.

3.3.1 Descriptive and Exploratory Analysis

Table 2.2 shows the statistics of the considered datasets. The datasets present almost the same distribution of subject characteristics. The box-plots in Figure 3.4 show more information about dispersion, position, and outliers of the subjects characteristics. There are many outliers for the variable Age which means that there are some person with age out of 1.5 interquartile difference. Despite of this fact, the variability is not low. For height and weight we observe more variability and few outliers.

An other descriptive statistics we performed is the frequency distribution of the activities. In Figure 3.5 we present activities distribution respect to the three datasets. As

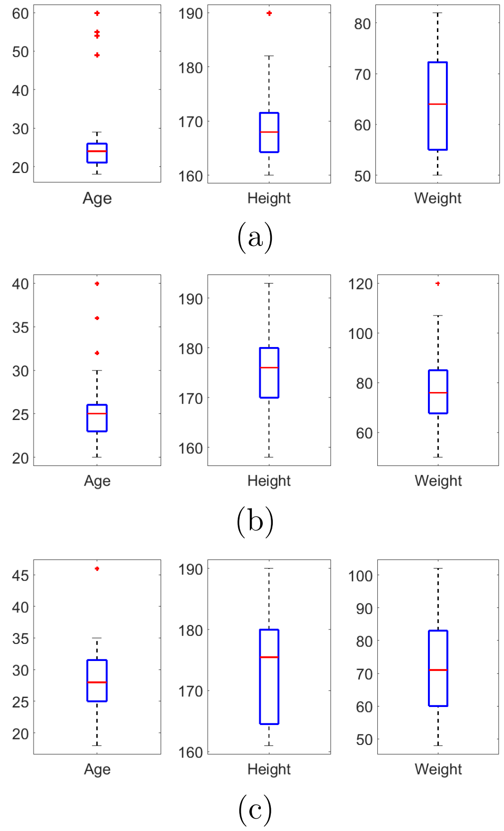


Figure 3.4: Boxplots over Age, Weight and Height: (a) UniMiB-SHAR; (b) MobiAct; (c) Motion Sense.

we can see in the graphs, UniMiB-SHAR and MobiAct contain more activities than Motion Sense and more important some activities, such as walking, standing, running and jogging, are more frequent than other activities such as stairs up, falling right, etc.

To further motivate our work we performed a multidimensional scaling analysis of the physical characteristics of subjects of each dataset [36]. This analysis may permit to highlight similarity between subjects that share similar physical characteristics. We applied the analysis to a matrix $D^{physical}$ of size $N \times N$ where N is the number of subjects of a given database. The matrix $D^{physical}$ is calculated by calculating the similarity between all the N subjects of the given database using equation (2) mentioned in section 3.2.2.

Figure 3.6 shows the results of such an analysis mapped into 2 dimensions. Each row of the figure is related to a given dataset. Each point of the graph represents one subject of the dataset and each color shows a given cluster of subjects.

For each dataset, in the figures on the left, red stands for young subjects and blue for older subjects, while in the figures on the right, green stands for normal Body Max Index (BMI) subjects and red stands for subjects with an abnormal BMI. The result of

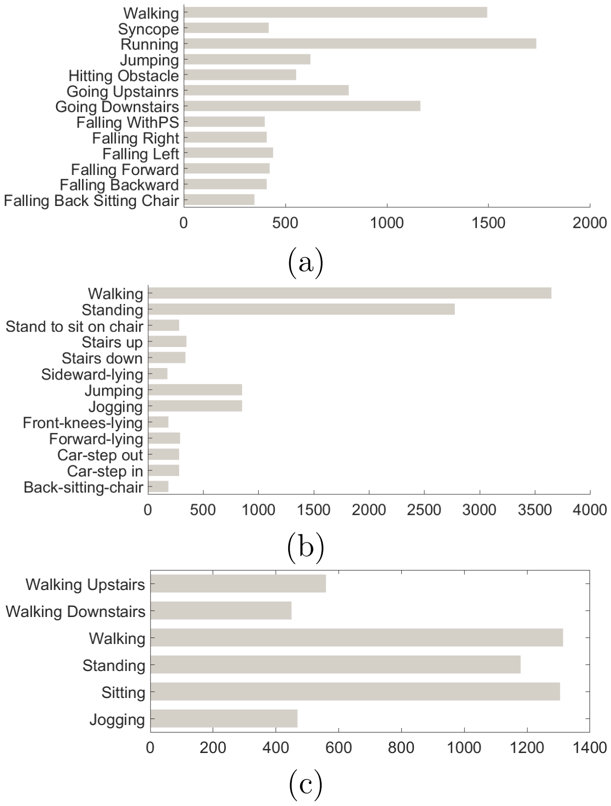
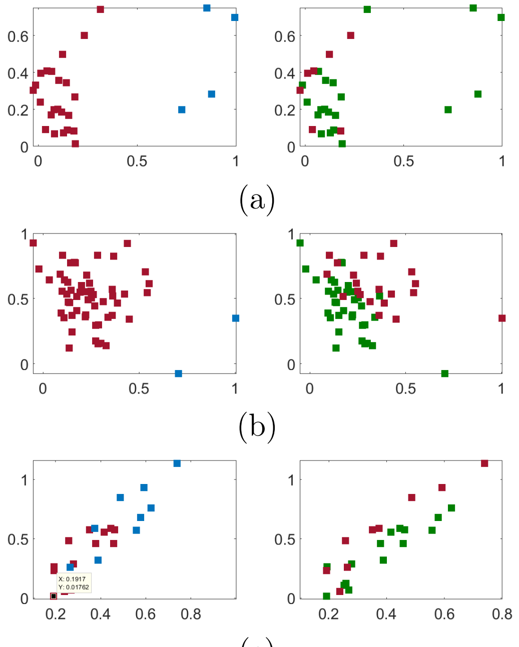


Figure 3.5: Activity sample Distribution: (a) UniMiB-SHAR; (b) MobiAct; (c) Motion Sense.

this analysis substantially confirms that a kind of implicit separability, on the basis of physical characteristics, between subjects exists.

The results of the multiscale analysis highlighted that age and physical characteristics are discriminator factors under subjects. In all the cases, age discriminates subjects, while BMI discriminates in two cases over three.

Until now we just explored datasets in terms of physical aspects. Here we want to investigate if signals present some discriminator tendency. For this reason we exploited the Principal Component Analysis (PCA) [66] over the features presented in Table 2.2. The first and second principal component have been computed for all datasets and are shown in Figure 3.7. The points on the graphs represent the subjects and the blue color stands for subjects with a $21 < BMI < 28$ and the red color stands for subjects with BMI values outside the interval considered above. PCA analysis shows evidently that subjects are separable on the basis of the features extracted from the segments of the original accelerometer signals.

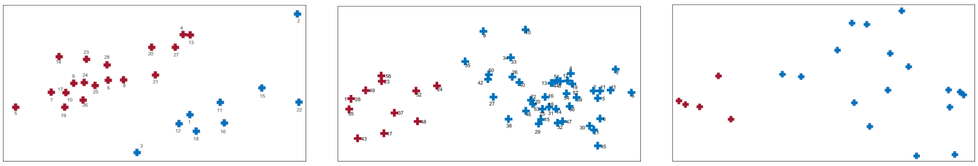


(a)

(b)

(c)

Figure 3.6: Multidimensional Scaling over physical characteristics. (a) UniMiB-SHAR; (b) MobiAct; (c) Motion Sense.



(a) (b) (c)

Figure 3.7: PCA using 18 features. (a) UniMiB-SHAR; (b) MobiAct; (c) Motion Sense.

3.4 Results

As discussed in section 3.2.3 we have performed 2 groups of experiments: 1) *experiments without similarity-based personalization* and 2) *experiments with similarity-based personalization.*

For sake of comparison we have experimented two different classification strategies for each group of experiments: AdaBoost combined with hand crafted (AdaBoost-HC), see Table 2.2, AdaBoost combined with deep features (AdaBoost-CNN). We have used the Matlab version of AdaBoost (MathWorks). To further evaluate the goodness of deep features, we experimented the use of Support Vector Machines (SVM) combined with deep features (SVM-CNN) in the case of the first group of experiments. We have used

Layer name	Shape
conv1	$\{1{\times}3\} \times n$
conv2_n	$\{1{\times}3 \times f_{maps}\} \times n$
conv3_n	$\{1{\times}3 \times 2f_{maps}\} \times n$
conv4_n	$\{1{\times}3 \times 4f_{maps}\} \times n$
avg_pool_x	$1{\times}32$
fully conn.	$(1 \times 4f_{maps}) \times 15$
softmax	$1{\times}15$

Table 3.2: Residual Network Architecture

the Matlab version of SVM (MathWorks).

Deep features have been achieved by using the Residual Network developed in [40]. Table 3.2 details the network architecture proposed for this study. The input size of the network is $1 \times 128 \times 3$, that corresponds to 3 segments along the three axes x, y, and z. The network architecture is made of an initial convolutional block, 3 residual stages, each containing a variable number n of residual blocks, average pooling layer, fully connected layer, and softmax layer. A convolutional block is made of three layers: convolutional, batch normalization, and ReLu. A residual block is made of 2 subsequent convolutional blocks and an addition operator that sums the input of the residual block with the output of the residual block itself. Each convolutional layer is $1 \times 3 \times f_{maps}$, where f_{maps} is the number of feature maps of the filter. The best values for n and f_{maps} have been found by following a grid search approach: n ranged between 3 and 21, while f_{maps} ranged between 10 and 200.

The network has been trained on the UCI-HAR [7] dataset and then used as feature extractor. In particular, we have taken the last pooling layer before the last fully connected layer thus obtaining a 1024-dimensional feature vector.

3.4.1 Experiments without similarity

For the first group of experiments, we have randomly initialized the weights of the algorithm while for the second group, we have weighted all the data belonging to a given subject with the corresponding weights given by the similarity between the given subject and the test subject.

Table 3.3 reports the results achieved by using the AdaBoost classifier for both the groups of experiments. The second group of experiments is highlighted using a light gray color while the remaining are numbers related to the first group of experiments. On average, looking at the fourth column of the first group of experiments, it is quite clear that the hybrid strategy works better than the subject-independent one, the improvement is of about 3% (see also Table 3.4). This behavior was some how expected, because the hybrid strategy considers the training set containing a small amount of data that belongs to the subjects under test. The presence of data of subject under test permits the AdaBoost to better specialize on the subject under test.

	UniMiB-SHAR	**MobiAct**	**Motion Sense**	**AVG**
subject-dependent	84.79	45.57	43.55	57.97

	UniMiB-SHAR	**MobiAct**	**Motion Sense**	**AVG**
	Δ max % ∼ 24	Δ max % ∼ 7	Δ max % ∼ 4	Δ max 14%
subject-independent	56.80	81.29	72.48	70.19
hybrid	61.66	83.73	73.82	73.07
subject-independent - Physical Similarity	57.39	81.62	72.45	70.49
subject-independent - Sensor Similarity	57.00	82.45	74.03	71.16
subject-independent - Physical Sensor Similarity	56.93	82.64	73.85	71.14
hybrid - Physical Similarity	**85.44**	89.43	77.76	84.21
hybrid - Sensor Similarity	84.71	90.76	**78.06**	**85.51**
hybrid - Physical Sensor Similarity	84.87	**90.90**	77.86	84.53

Table 3.3: % Accuracy of the first and second group (light gray) of experiments achieved using AdaBoost-HC.

	AdaBoost-HC	**AdaBoost-CNN**	**SVM-CNN**	**AVG**
	Δ max % ∼ 14	Δ max % ∼ 30	–	Δ max % ∼ 24
subject-dependent	57.97	62.32	86.01	68.76
subject-independent	70.19	60.61	60.60	63.80
hybrid	73.07	72.36	70.23	71.89
subject-independent - Similarity	70.93	61.67	-	66.30
hybrid - Similarity	**84.75**	**90.03**	-	**87.39**

Table 3.4: % Accuracy of the first and second group (light gray) of experiments using AdaBoost-HC, AdaBoost-CNN and SVM-CNN.

Moreover, apart from UniMiB-SHAR, the subject-dependent strategy achieves an average accuracy of about 35% lower than both subject-independent and hybrid strategies. In the case of UniMiB-SHAR, this is not true because the dataset is made of segments taken around peaks of the accelerometer signal while the other datasets are made of segments taken subsequently with a zero or 50% of overlap. In case of walking, there is an high probability that in a segment of 3 seconds there are many peaks (higher than 2g). Let us suppose that the number of peaks is 5, it means that for a segment of 3 seconds we take 5 segments, that are quite similar, for classification. The resulting dataset contains a large amount of redundant segments. A subject-dependent strategy takes advantage of this redundancy and specializes very well the classifier especially when the training set is made of only data from the subject under test. In the case of MobiAct and Motion Sense datasets, this is not true because we take a segment of 5 seconds that shares 50% of its length with the previous and subsequent segments.

The comparison between AdaBoost-HC, AdaBoost-CNN, and SVM-CNN is showed in Table 3.4. Results are averaged across datasets. Numbers show that overall, the use of hand-crafted features outperform the use of CNN features. However, the best performance is achieved by the Adaboost-CNN combination with hybrid model and similarity. This behavior is also confirmed by the results of the SVM-CNN approach that, apart from the subject-dependent case, are quite similar to the results achieved by the AdaBoost-CNN approach.

3.4.2 Experiments with similarity

The three similarity-based personalizations are then computed on the basis of physical characteristics, signal characteristics, and physical combined with signal characteristics. These personalizations have been applied to both subject-independent and hybrid strategies. The corresponding results for AdaBoost-HC are highlighted in Table 3.3 with a light gray color. On average, looking at the fourth column of the group 2 of experiments, it is quite clear that the similarity-based personalizations lead to a considerable improvement only in the case of the hybrid strategy. In the case of UniMiB-SHAR, the maximum improvement that we achieved is of about 0.5% and 24% for subject-independent and hybrid strategy respectively. In the case of MobiAct, the maximum improvement is of about 1.5% and 7% for subject-independent and hybrid strategy respectively. In the case of Motion Sense, it is of about 1.5% and 4% for subject-independent and hybrid strategy respectively. In Table 3.4 results achieved by the AdaBoost-CNN approach confirm that the use of similarity increase the performance with a highest improvement of about 30%. Across datasets, on average, similarity-based personalizations lead to an improvement of performance of about 0.9% and 14.7% for subject-independent and hybrid strategy respectively (see also Table. 3.5).

The fact that similarity-based personalizations combined with the hybrid strategy work better than personalizations combined with the subject-independent strategy is not surprising. Similarities between subjects are used to weight the data of the training set. In practice data belonging to more similar subjects are more important than data belonging to less similar subjects.

Among similarity-based personalizations, differences are few. It is clear from numbers that physical, signal, and physical + signal -based similarities lead to almost the same improvement of accuracy.

3.5 Conclusions

Recently, a significant amount of literature concerning machine learning techniques has focused on human automatic activity recognition (HAR) by using accelerometer recorded by smartphones. Real-world HAR systems may achieve not satisfying recognition accuracy

	no-similarity	similarity	Δ%
subject-dependent	60.15	-	-
subject-independent	65.40	66.30	0.9
hybrid	72.72	87.39	14.7

Table 3.5: % Average of accuracy achieved on the subject-dependent split and on subject-independent and hybrid combined with similarity-based personalization methods.

in real world applications because HAR techniques may struggle to generalize to new users and/or new environments. Several factors may affect the accuracy of activity recognition methods: i) position of the device; ii) differences between different brands of sensors; iii) human characteristics. While factors related to the position and the characteristics of the devices have been largely investigated, few works have explored the effects of human characteristics on recognition accuracy.

In this Section we presented several personalization methods on three public datasets in order to make the results reproducible and thus allowing future research on this topic. The personalization methods experimented are based on the concept of similarity between users. This means that users may have similar physical characteristics or have similar accelerometer signals and that, such a similarity can be employed to weight training data in a way that data belonging to more similar subjects to the subject under test count more than data of less similar subjects.

We have combined personalization methods with suitable splits of the data: subject-independent and hybrid. The first split considers training set made of data from all the subjects but the subject under test, while the second split considers a training set made of data from all the subjects but the user under test and a small amount of data of the user under test. Experiments, on average, prove that personalization methods improve accuracy of the classifier only if combined with a hybrid split. In this case the increment of accuracy, on average, is of about 11%.

Results in this section confirm that personalization methods can be effective especially if a small amount of subject-dependent data are included in the training set. The way we carried out the experimentation makes it possible to reproduce the results and more important it paves the way for future investigations on this topic.

Chapter 4

Personalization in HAR: How Far Can We Go With Deep Learning?

4.1 Motivation

During the last decade, a plenty of traditional machine learning as well as deep learning methods have been proposed in literature [70, 95, 144, 107, 138]. In the recent literature, deep learning methods are predominant [138]. Deep learning methods require a special hardware setup (Graphical Processing Units - GPUs) to speed up computation and a great amount of data to avoid overfitting during the training process. However, it is very rare to find consumer hardware equipped with GPUs, thus in most cases, deep learning methods run on cloud platform, such as, Google Cloud[1], Amazon AWS[2], and Microsoft Azure[3]. Large scale inertial datasets with millions of signals recorded by hundreds of human subjects are still not available, but several smaller datasets made of thousands of signals and dozens of human subjects are publicly available [117]. A recent platform to support long-term data collection of inertial signals have been proposed [39, 52] with the scope to make available to the scientific community a large dataset enriched with context information (e.g., characteristics of the subject, device position etc.). Moreover, since the public available datasets for HAR benchmarking are not consistent, both syntactically (e.g., different sampling frequency) and semantically (e.g., labels with different meanings), Ferrari et al. have proposed a platform for data integration [39] as well as methods for data homogenization [41].

[1] https://cloud.google.com

[2] https://aws.amazon.com/

[3] https://azure.microsoft.com

Since to date, large scale inertial datasets are not available, it is therefore not obvious in this domain, which method between deep and traditional machine learning methods is the most appropriate, especially in those case where the hardware is low cost.

This Chapter aims at comparing deep learning techniques with traditional ones, also considering personalization. In particular, in Section 4.2, traditional deep learning and traditional machine learning are compared; in Section 4.3, deep learning and personalized machine learning are compared. Finally, in Section 4.4, personalized deep learning and personalized machine learning are evaluated.

4.2 Does Deep Learning outperform Traditional Machine Learning techniques?

The aim of this Section is to investigate the robustness of traditional classifiers combined with hand-crafted features compared with an end-to-end deep learning solution based on a Residual Network that is one of the most performing network in the state of the art [19]. In particular, deep learning benchmark methods are evaluated using the acceleration, the gyroscope and both. Experiments on four public datasets are presented and discussed. In Section 4.2.1 we select state-of-the-art machine learning and deep learning models for evaluation. We briefly describe different features extraction procedure as data input for machine learning approaches. In Section 4.2.2 experimental setup is presented. In Section 4.2.3 results for single and multi-modality are discussed. Section 4.2.4 the conclusions.

4.2.1 Proposed Methods

In this Section, benchmark machine learning and deep learning methods are compared with different data input. In particular, raw data, and hand crafted features have been calculated from acceleration and gyroscope data.

4.2.1.1 Hand-crafted features

For the experimentation of hand-crafted features, the k Nearest Neighbour (k-NN) and Support Vector Machines (SVM) classifiers have been used. The features used are:

- Raw data (denoted as *raw*): x,y, and z accelerometer segments (without any kind of processing) are concatenated and used as feature vectors [89];

- Magnitude of the segments (denoted as *magn*) [88];

- 20 features extracted from the magnitude of the segments (denoted as *hc magn*) [20]. Table 2.2 reports details about the 21 features.

- 20 features extracted from each of the three segments along the three axes x, y, and z (denoted as *hc raw*). The total number of features is 63.

In the case of SVM, the multi-class classifier has been implemented as multiple binary classifiers. Optimum parameters of both classifiers have been found through cross-validation.

4.2.1.2 End-to-end deep learning solution

The Residual Network (ResNet) adopted for this study is based on the traditional architecture proposed by He *et al.* [59], which demonstrated to be very effective on the ILSVRC 2015 (ImageNet Large Scale Visual Recognition Challenge) validation set with a top 1-recognition accuracy of about 80%. Residual architectures are based on the idea that each layer of the network learns residual functions with reference to the layer inputs instead of learning unreferenced functions. He *et al.* [59] demonstrate that such architectures is easier to optimize and it gains accuracy also when the depth increase considerably.

Table 3.2 details the network architecture proposed for this study. The input size of the network is $1 \times 128 \times 3$, that corresponds to 3 segments along the three axes x, y, and z. The network architecture is made of an initial convolutional block, 3 residual stages, each containing a variable number n of residual blocks, average pooling layer, fully connected layer, and softmax layer. A convolutional block is made of three layers: convolutional, batch normalization, and ReLu. A residual block is made of 2 subsequent convolutional blocks and an addition operator that sums the input of the residual block with the output of the residual block itself. Each convolutional layer is $1 \times 3 \times f_{maps}$, where f_{maps} is the number of feature maps of the filter. For each dataset, the best values for n and f_{maps} have been found by following a grid search approach: n ranged between 3 and 21, while f_{maps} ranged between 10 and 200.

Figure 2.11 shows the best network for UCI-HAR, obtained with $n = 1$ and $f_{maps} = 90$. For all the datasets, the networks have been optimized through the Stochastic Gradient Descent with Momentum (SGDM), using a piecewise learning update strategy with an initial value of 0.1 and a drop factor of 0.1. The batch size was 128, the total number of epochs was 80 and the early stopping has been used to avoid overfitting.

4.2.2 Experiment setup

Four public datasets from Table 2.1 have been used for the analysis:

- *UCI HAR* [7], which includes tri-axial acceleration and gyroscope data of 6 ADLs (Activities of Daily Living) recorded with a Samsung Galaxy S II and performed by 30 volunteers.

- *MobiAct* [135], which includes tri-axial acceleration, gyroscope, and orientation data of 11 ADLs and 4 Falls recorded with a Samsung Galaxy S3 and performed by 67 volunteers.

Dataset	# train	# validation	# test	# classes
UCI HAR	7209	2060	1030	6
MobiAct	34070	9734	4867	15
Motion Sense	14945	4270	2135	6
UniMiB-SHAR	8240	2354	1177	17

Table 4.1: Number of segments divided into training, validation and test dataset and the number of the classes for each dataset.

- *Motion Sense* [86], which includes tri-axial acceleration and gyroscope data of 6 ADLs recorded with an iPhone 6s and performed by 30 volunteers.

- *UniMiB-SHAR* [88], which includes tri-axial acceleration data of 17 ADLs recorded with an Samsung Galaxy Nexus I9250 and performed by 24 volunteers.

Considering the acceleration only, each signal of the datasets is composed of three accelerometer components along the x, y, and z axis. Each signal component has been resampled at 50Hz and divided in segments of 2.56 seconds with an overlap between subsequent segments of 50% [107]. The resampling at 50Hz was necessary because the MobiAct dataset has been acquired at a frequency of about 87Hz. The resulting segment for each axis contains 128 samples. The resampling and segmentation procedures were not applied to UniMiB-SHAR because such a dataset already contains overlapped segments of 151 samples. In fact, the dataset contains segments of 3 seconds sampled at 50Hz taken around a peak (higher than 1.5g, with g being the gravitational acceleration) of the accelerometer signal. To be consistent with the other datasets, these segments were centrally windowed in order to obtain 128-dimensional segments.
Each dataset has been split in 70% training, 20% validation, and 10% test. Table 4.1 shows the total number of 128×3-dimensional segments available for the training (column *# train*), validation (column *# validation*), and test (column *# test*) sets. The last column *# classes* indicates the number of ADLs present in the dataset.

UCI HAR, MobiAct and Motion Sense have been analyzed taking into account the accelerometer and the gyroscope sensors, while UniMiB-SHAR has been exploited only for the accelerometer sensor because of the lack of gyroscope data.

4.2.3 Results

4.2.3.1 Results for Single Modality

Tables 4.2 and 4.3 show results achieved by all the methods considered in terms of macro average accuracy (i.e., the average of each class accuracy). The accuracy of each class is computed as ratio between the number of segments correctly classified and the total number of segments of that class. ResNet achieves better performance than traditional methods in all datasets apart from MobiAct. Most important, the standard deviation of

	SVM				**ResNet**
Dataset	raw	magn	hc raw	hc magn	
UCI-HAR	79.51 (± 17.40)	53.10 (± 25.48)	79.47 (± 20.59)	48.45 (± 22.12)	**90.73 (± 10.92)**
MobiAct	77.93 (± 22.71)	63.63 (± 24.13)	76.73 (± 26.11)	59.95 (± 23.94)	**92.98 (± 8.65)**
Motion Sense	90.04 (± 14.36)	78.22 (± 29.59)	96.39 (± 3.79)	83.45 (± 21.13)	**99.47 (± 0.87)**
UniMiB-SHAR	58.26 (± 16.85)	52.27 (± 18.10)	58.08 (± 16.70)	50.81 (± 15.49)	**88.59 (± 8.52)**

Table 4.2: Experimental Results - mean class accuracy (standard deviation class accuracy) with different input data: raw, magnitude (magn), hand-crafted calculated on raw data (hc-raw) and hand-crafted on magnitude (hc-magn). Comparison between SVM vs ResNet.

	***k*-NN**				**ResNet**
Dataset	raw	magn	hc raw	hc magn	
UCI-HAR	73.71 (± 26.78)	46.92 (± 29.89)	69.35 (± 17.04)	37.75 (± 13.39)	**90.73 (± 10.92)**
MobiAct	87.69 (± 9.07)	77.81 (± 13.60)	91.86 (± 6.72)	80.50 (± 10.74)	**92.98 (± 8.65)**
Motion Sense	79.19 (± 31.83)	73.51 (± 25.16)	95.82 (± 5.61)	81.34 (± 20.30)	**99.47 (± 0.87)**
UniMiB-SHAR	61.97 (± 11.83)	55.13 (± 14.29)	65.74 (± 12.99)	52.22 (± 11.70)	**88.59 (± 8.52)**

Table 4.3: Experimental Results - mean class accuracy(standard deviation class accuracy)with different input data: raw, magnitude (magn), hand-crafted calculated on raw data (hc-raw) and hand-crafted on magnitude (hc-magn). Comparison between *k*-NN vs ResNet.

the ResNet method is close to zero. Accuracy of *k*-NN and SVM is quite similar, while among hand-crafted features the most performing is the *hc raw*.

Overall, ResNet is the best performing with an average accuracy across datasets of 92.94%, the second best across classifiers and datasets are the *hc raw* features with an average accuracy of 79.18%. The third best are the *raw* features with an average accuracy of 76.04%. The worst are the *magnitude* and *magnitude raw* features with an average accuracy of 62.57% and 61.81% respectively.

In summary, the average gap between hand-crafted features combined with traditional classifiers and deep learning is about 15%. This experimentation actually confirms that deep learning outperforms traditional machine learning approaches.

4.2.3.2 Results for Multimodality

Tables 4.4 and 4.5 show results achieved by all the methods considered in terms of macro average accuracy (i.e., the average of each class accuracy). The accuracy of each class is computed as ratio between the number of segments correctly classified and the total number of segments of the given class. ResNet achieves better performance than traditional methods in all datasets apart from MobiAct. This is probably due to the larger number of classes with respect to the other two datasets. Most important, the standard deviation of the ResNet method is close to zero. Accuracy of *k*-NN and SVM is quite similar, while, among hand-crafted features, the most performing is the *hc raw*.

		KNN				**ResNet**
	Dataset	raw	magn	hc raw	hc magn	
ACC	UCI-HAR	73.71 (± 26.78)	46.92 (± 29.89)	69.35 (± 17.04)	37.75 (± 13.39)	**90.73 (± 10.92)**
	MobiAct	87.69 (± 9.07)	77.81 (± 13.60)	91.86 (± 6.72)	80.50 (± 10.74)	**92.98 (± 8.65)**
	Motion Sense	79.19 (± 31.83)	73.51 (± 25.16)	95.82 (± 5.61)	81.34 (± 20.30)	**99.47 (± 0.87)**
GYRO	UCI-HAR	70.74 (± 24.73)	38.37 (± 25.11)	60.19 (± 20.14)	33.50 (± 6.97)	**89.36 (± 9.90)**
	MobiAct	78.54 (± 16.13)	73.66 (± 19.38)	83.19 (± 20.43)	74.62 (± 20.44)	**96.09 (± 3.16)**
	Motion Sense	85.16 (± 12.03)	70.75 (± 22.24)	88.74 (± 10.31)	71.69 (± 13.62)	**98.07 (± 1.56)**
ACC+GYRO	UCI-HAR	82.36 (± 19.99)	55.35 (± 29.60)	77.10 (± 16.73)	39.74 (± 11.79)	**96.46 (± 4.06)**
	MobiAct	86.25 (± 8.89)	80.22 (± 14.85)	**94.20 (± 5.84)**	81.46 (± 11.59)	92.94 (± 9.39)
	Motion Sense	74.08 (± 29.86)	73.89 (± 24.70)	97.17 (± 2.33)	84.36 (± 9.45)	**99.08 (± 0.65)**

Table 4.4: Experimental Results - average accuracy (standard deviation) with different input data: raw, magnitude (magn), hand-crafted calculated on raw data (hc-raw) and hand-crafted on magnitude (hc-magn). Comparison between k-NN vs ResNet. For each row, the bold font represents the best.

Figure 4.1(a) shows the comparison between ResNet and both k-NN and SVM across dataset and independently of the inertial signal used. Overall, ResNet is the best performing with an average accuracy across datasets of about 93%. Among the traditional classifiers, the best results are achieved by *hc raw* features followed by the *raw* ones. The worst results are obtained by using magnitudes (both *hc magnitude* and *magnitude raw*).

Figure 4.1(b) shows the comparison across datasets, between methods based on sensorial multimodality (i.e., accelerometer and gyroscope - ACC+GYRO) and single modality (i.e., accelerometer or gyroscope - ACC or GYRO). Overall, multimodal recognition works better than the single modal with an improvement of about 10%. Accelerometer is more performing than the gyroscope. This result is confirmed by the fact that most of the experiments undertaken in the literature are based on accelerometric signals only.

In summary, the average gap between hand-crafted features combined with traditional classifiers and deep learning is about 10% thus confirming that, on these datasets, deep learning approaches outperforms traditional ones.

4.2.4 Conclusions

Experiments on four public datasets demonstrated that overall deep learning solutions overcome the state of the art, thus suggesting that, even when the large scale datasets are not available, these techniques on average perform better than traditional machine learning approaches. The joint use of accelerometer and gyroscope allows to increase performance of about 10% with respect to the use of accelerometer or gyroscope alone. However, hand-crafted features may be preferable in those cases where the hardware is low cost and thus does not permit deep learning solutions to run in a short time.
Based on these results, which show the benefit in using deep learning for HAR, we want

		SVM				ResNet
	Dataset	raw	magn	hc raw	hc magn	
ACC	UCI-HAR	79.51 (± 17.40)	53.10 (± 25.48)	79.47 (± 20.59)	48.45 (± 22.12)	**90.73 (± 10.92)**
	MobiAct	77.93 (± 22.71)	63.63 (± 24.13)	76.73 (± 26.11)	59.95 (± 23.94)	92.98 (± 8.65)
	Motion Sense	90.04 (± 14.36)	78.22 (± 29.59)	96.39 (± 3.79)	83.45 (± 21.13)	**99.47 (± 0.87)**
GYRO	UCI-HAR	72.93 (± 23.82)	44.52 (± 27.21)	75.45 (± 14.76)	41.10 (± 17.91)	**89.36 (± 9.90)**
	MobiAct	64.19 (± 31.37)	57.80 (± 27.49)	68.86 (± 27.70)	52.21 (± 29.22)	**96.09 (± 3.16)**
	Motion Sense	86.92 (± 7.51)	73.93 (± 21.86)	88.32 (± 9.86)	76.46 (± 13.17)	**98.07 (± 1.56)**
ACC+GYRO	UCI-HAR	86.83 (± 15.53)	59.49 (± 28.25)	88.14 (± 10.66)	49.20 (± 20.45)	**96.46 (± 4.06)**
	MobiAct	79.13 (± 18.25)	70.15 (± 22.94)	85.54 (± 16.31)	62.77 (± 23.99)	**92.94 (± 9.39)**
	Motion Sense	85.87 (± 8.05)	80.93 (± 11.55)	95.90 (± 3.07)	85.01 (± 9.29)	**99.08 (± 0.65)**

Table 4.5: Experimental Results - average accuracy (standard deviation) with different input data: raw, magnitude (magn), hand-crafted calculated on raw data (hc-raw) and hand-crafted on magnitude (hc-magn). Comparison between SVM vs ResNet. For each row, the bold font represents the best.

to investigate further the property of robustness of these techniques. The goal of the next Section 4.3 is to investigate whether end-to-end deep learning methods also outperform personalized machine learning techniques.

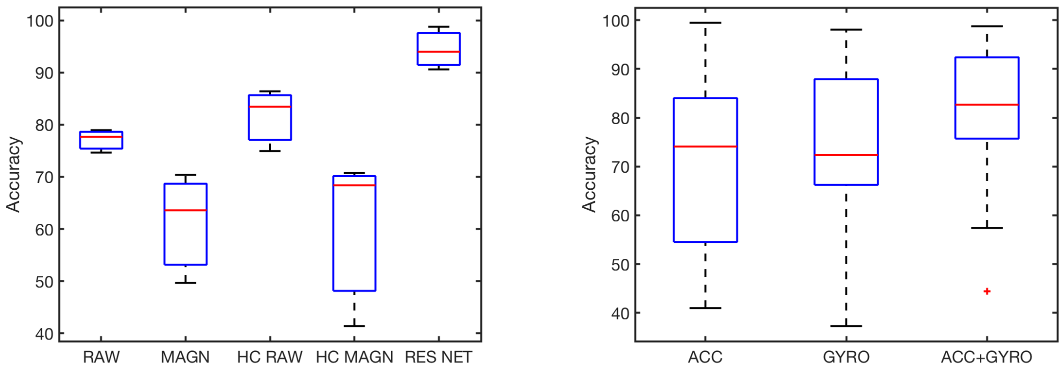
4.3 Does Deep Learning outperform Personalized Machine Learning?

Experiments in Chapter 3 and in Section 4.2 show that personalized machine learning and deep learning outperform traditional machine learning methods. Indeed, personalized machine learning improves overall performance when similarity between users are involved into the classification procedure, while deep learning is very robust although large scale datasets are not available and outperform over SVM and *k*-NN algorithms even with different input features.

In this Section we focus on the comparison between deep learning and personalized machine learning methods based on the public datasets, UniMiB-SHAR, MobiAct and Motion Sense. The comparison aims at investigating the robustness of the deep learning techniques in terms of *intra* and *inter* variability across subject, explained in 2.3.

We exploit transfer learning method for deep learning in order to evaluate deep learning performances in a low time consuming configuration. In Section 4.3.1 we briefly present the personalized machine learning models, for more details see Chapter 3, and the end-to-end deep learning models based on Convolutional Neural Network.

In Section 4.3.2 we present details about the datasets and the configuration of the final



(a) (b)

Figure 4.1: Experiments. (a) comparison across datasets between hand-crafted and ResNet. (b) comparison across datasets and methods between multimodality and single modality.

input data. In Section 4.3.3 the result and the comparison between the above mentioned techniques. Finally, section 4.3.4 provides final remarks.

4.3.1 Proposed Methods

4.3.1.1 Personalized machine learning models

Traditional machine learning techniques struggle to recognize new unseen user because of the *population diversity*, largely discussed in Chapter 3. To take into account such a diversity, we introduce the concept of similarity between subjects. Three types of similarity have been defined: *physical-based* ($\mathrm{sim}^{physical}$), *sensor-based* (sim^{sensor}), and *physical combined with sensor-based similarity* ($\mathrm{sim}^{physical+sensor}$). The similarity have been used to create a weighted classifier able to give more importance to data belonging to most similar subjects. As shown in Chapter 3, machine learning models with similarity outperform machine learning models without similarity.
Data have been pre-processed and split into subject - dependent,subject - independent and hybrid. The classification has been lead with Adaboost ensemble classifier.

4.3.1.2 End-to-end Deep learning models

Table 4.6 details the network proposed for transfer learning. The input size is $1 \times 150 \times 3$. The network is composed by a convolutional layers with filter size equal to 3, an activation layer with the ReLu activation function, one max pooling layer with size equal to 3, the dropout layer with a dropout probability equal to 0.9, a fully connected layer, and the softmax layer.

For transfer learning we use a pre-trained network on the UCI-HAR dataset. Experiments have been evaluated on data splits subject - dependent, subject - independent and hybrid. None of the subject similarities mentioned above have been taken in consideration.

Layer name	**Shape**
convolutional	$\{1 \times 3\}$
activation	ReLU
max pooling	1×3
dropout	0.9
fully connected	$148 \times f_{maps}$
softmax	$1\times$ num classes

Table 4.6: Convolutional Neural Network Architecture

4.3.2 Experiment setup

We experimented three public datasets containing accelerometer signals of Activities of Daily Living (ADLs) and Falls recorded the smartphones presented in section 3.3. That is, UniMiB-SHAR, MobiAct and Motion Sense datasets have been pre-processed as explained in section 3.2.1.3.

4.3.3 Results

In this section, we discuss and compare the results of personalized machine learning (PML) and deep learning models (DL) evaluated on subject - independent, subject - dependent, and hybrid data split.

Tables 4.7 and 4.8 show results achieved by the proposed models. Results regarding the three datasets are shown in column. Results subdivided with different data splits are organized in different rows. In particular, Table 4.7 reports deep learning (DL) models and personalized machine learning accuracy in the subject-independent and hybrid case. Results under DL are calculated as an overall average over subjects and k-fold cross-validation. Personalized machine learning (PML) accuracy is calculated averaging results from Table 3.3 over the similarity, i.e. physical, sensors, and physical+sensors.

The hybrid model presents, in general, an accuracy of about 14% and 6% higher for, respectively, PML and DL. In MobiAct and Motion Sense datasets best accuracy is achieved by deep learning strategies for both subject independent and hybrid models. Concerning the UniMiB-SHAR dataset, the PML with hybrid achieves the best accuracy. The former result makes personalized machine learning method in average preferable with hybrid strategy, with an average accuracy of the 84.42%. In subject-independent data split, DL achieved an accuracy of about 6% higher then PML.

	UniMiB-SHAR	**MobiAct**	**Motion Sense**	**average**
	PML - DL	**PML - DL**	**PML - DL**	**PML - DL**
subject-independent - similarity	57.11 - 58.88	82.24 - 88.92	73.44 - 81.03	70.93 - **76.28**
hybrid - similarity	85.00 - 69.71	90.36 - 92.62	77.89 - 85.75	**84.42** - 82.69
average	**71.05** - 64.3	86.3 - **90.77**	75.67 - **83.39**	

Table 4.7: Experimental Results - accuracy of personalized machine learning (PML) vs accuracy of traditional deep learning (DL) on subject - independent and hybrid data splits. For each row, the bold font represents the best results.

	UniMiB-SHAR	**MobiAct**	**Motion Sense**	**average**
	PML - DL	**PML - DL**	**PML - DL**	**PML - DL**
subject dependent	**84.79** - 78.77	45.57 - **82.90**	43.55 - **82.46**	57.97 - **81.38**

Table 4.8: Experimental Results - accuracy of personalized machine learning (PML) vs accuracy of traditional deep learning (DL) on subject - dependent data splits. For each row, the bold font represents the best.

The same behavior is presented in Table 4.8. UniMiB-SHAR achieve highest performance by using PML, while for MobiAct and Motion Sense DL remains preferable with a difference of about 40% to PML. In average, DL outperforms PML methods.

4.3.4 Conclusions

Promising results highlighted in Section 4.3.3 and obtained in Chapter 3 led us to compare personalized machine learning with deep learning methods. Indeed, both personalized machine learning and deep learning methods have shown higher performance compared to traditional machine learning methods. In particular, deep learning has shown to be a valid strategy to overcome the domain expert dependency of machine leaning methods on the features extraction procedure. Nevertheless, deep learning techniques are expensive in terms of time consuming which makes machine learning techniques be preferred in many contexts.

In this Chapter we compared personalized machine learning and deep learning techniques. In particular, we use transfer learning for deep learning strategy to the aim at reducing time's consume.

Results show that deep learning accuracy outperforms personalized machine learning accuracy in most of the cases also using transfer learning strategy and without any fine tuning procedure. In contrast, results on UniMiB-SHAR dataset show better accuracy using hybrid personalized machine learning approach.

Given the achieved results, it is still difficult to state that there exists a best classifier for HAR, which motivates us to further try to modify, improve and compare machine learning deep learning techniques. In the next section 4.4, we investigate if the similarity between users can improve results of deep learning techniques. In these terms, we explore the possibility to personalize deep learning algorithms in order to achieve overall better accuracy.

4.4 Does Personalized Deep Learning outperform Personalized and Traditional Machine Learning?

The results achieved with the experiments described in Section 4.2 and in Section 4.3, we confirm that deep learning is a powerful technique for HAR. In particular, deep learning techniques based on raw data outperform traditional machine learning techniques with different features input. Robustness in these terms permit to remove the dependency of the algorithm on the expert knowledge, which normally affects machine learning techniques and leads to a lack of generalization. Furthermore, higher accuracy makes deep learning techniques valid candidate to face *intra* and *inter* subject variabilities.

In Section 4.3 we showed that deep learning outperformed personalized machine learning in most of the cases as a consequence of their high robustness in terms of subjects variability. Nevertheless, it remains some cases where personalized machine learning techniques outperform deep learning, which motivates us to further investigate how to improve deep learning models.

The general idea is to create personalized deep learning models. In the literature, personalization in deep learning models is achieved using transfer learning and incremental learning. Both techniques rely on updates of an existing network with the data of the new user. User's similarity, such as physical attributes or sensor's data proximity are not considered.

We have defined a novel deep learning model based on Convolutional Neural Networks combined with the information on user's physical, and sensors similarities. The results have been compared with the personalized machine learning models presented in Chapter 3.

4.4.1 Proposed Methods

In this section, we describe the personalized deep learning models based on Convolutional Neural Network and on the similarity between the subjects.

The concept of subjects similarity is analogous to the one explained in Chapter 3. We considered three types of similarity: physical, sensor, and physical combined with sensor.

The similarity is defined as a function of the distance between vectors of two subjects. Thus, the physical similarity sim^{physical} is a function of the distance vector composed on subject's physical characteristics, namely the high, the weight, the age. The sensor similarity sim^{sensor} is a function of the distance between two signals vectors of two different subjects, and *physical combined with sensor-based similarity* sim$^{physical+sensor}$ based on the distance between both physical and sensors vector.

As remind, the similarity computed on all subjects generates a matrix, called similarity matrix sim. Each element $sim(i, j)$ represents the similarity between subjects i and j. Each value is between 0 and 1: 0 means that the two subjects are dissimilar, and 1 means that the two subjects are equal.

In the personalized machine learning methods we used these values as weights to feed the classifier together with the data.

In personalized deep learning models, the matrix is used in a different way, as described in the following.. Starting from a minimum value m we select the most m similar subjects, with respect to the test subject. The network is trained with the samples related to these m subjects. We selected as starting value for m the value 10, trained the network, and added 5 subjects until the maximum number of subjects in the dataset is achieved. This configuration is repeated for subject - independent and hybrid data splits.

In general, we believe that considering the most similar subjects leads to better performance of the algorithm. The classification should indeed not influenced from dissimilar subjects or outliers.

4.4.2 Experiment setup

To compare personalized deep learning results with personalized machine learning methods we propose again the same experimental setup proposed in Chapter 3. Three public datasets containing accelerometer signals of Activities of Daily Living (ADLs) and Falls recorded the smartphones have been considered. That is, UniMiB-SHAR, MobiAct and Motion Sense datasets have been pre-processed as explained in section 3.2.1.3. The details are omitted in this section because already described in Section 4.2.2.

4.4.3 Results

Table 4.9 shows the results of personalized deep learning methods (PDL) compared with personalized machine learning (PML) methods for each dataset. The number m represent the number of the most similar subjects compared to the test subject. The similarity matrix have been computed for all similarities, i.e. physical, sensor, and physical combined with sensor attributes. Table 4.9 show the accuracy achieved by the algorithm tested on different datasets, data splits and similarities. PDL with different choices of m and PML results are organized in columns. For the sake of clarity, MobiAct dataset's results referred to m = 30, 35, 40, 45, 50, 55 have been grouped together and the maximum and minimum accuracy are shown (PDL >=30).

Dataset	**Models**			**PDL**			**PML**
m-th nearest subjects		10	15	20	25	>=30 min - max	57
MobiAct	subject - independent- physical	75.88	78.96	81.54	82.59	83.02 - **86.08**	81.62
	subject - independent- sensor	71.75	74.36	76.25	77.68	77.42 - 80.14	**83.45**
	subject - independent- physical - sensor	71.88	74.11	75.97	77.38	78.45 - 79.68	**82.64**
	hybrid - physical	75.75	76.51	77.67	78.77	79.43 - 81.04	**89.43**
	hybrid - sensor	78.15	78.77	79.45	80.39	80.90 - 81.40	**90.76**
	hybrid - physical - sensor	85.23	86.32	86.96	87.40	87.58 - 88.17	**90.90**
average						82.75	**86.46**
m-th nearest subjects		10	15	20	25		27
UniMiB-SHAR	subject - independent- physical	25.49	27.61	31.48	35.42	-	**57.39**
	subject - independent- sensor	40.71	42.14	42.65	42.83		**57.00**
	subject - independent- physical - sensor	41.02	42.21	42.50	42.66		**56.93**
	hybrid - physical	42.87	43.69	45.33	45.82		**85.44**
	hybrid - sensor	47.26	45.99	46.77	46.49		**84.71**
	hybrid - physical - sensor	46.17	46.77	46.77	45.39		**84.87**
average					43.46		**71.05**
m-th nearest subjects		10	15	20			22
Motion Sense	subject - independent- physical	74.30	77.40	**78.02**	-	-	72.45
	subject - independent- sensor	75.91	77.83	**78.80**			74.03
	subject - independent- physical - sensor	75.77	77.76	**79.00**			73.85
	hybrid - physical	77.59	79.44	**80.17**			77.76
	hybrid - sensor	78.51	80.08	**80.38**			78.06
	hybrid - physical - sensor	78.79	80.25	**80.41**			77.86
average				**79.46**			75.66

Table 4.9: Experimental Results - accuracy of personalized deep learning (PDL) compared with personalized machine learning (PML).

PDL and PML's accuracy is calculated as the average over the subjects. In the case of PML, the number m is the total number of the subjects in the dataset, see Section 4.4.2 for details about datasets size.

In the following we discuss and compare the results in Table 4.9. For the sake of clarity, we first compare of the *PDL performances between the datasets* and second we discuss and compare the *performances across PDL with PML.*

- *PDL performance between the datasets*: in general, the MobiAct dataset achieves better performance using PDL models in comparison with the other datasets. The datasets size remains, in general, a crucial factor to determine the performance of the algorithm. The larger the training dataset, the better the performance. In particular, MobiAct training dataset size is of 12400 samples. UniMiB-SHAR has up to 6800 samples for training and Motion Sense up to 8000.

However, we observe that for a given m MobiAct has in general less training samples in comparison with UniMiB-SHAR and Motion Sense. In particular, the number of samples and the related accuracy are as follows

- m = 10, MobiAct has 2146 training samples with an accuracy of 85.23%, UnIMiB-SHAR has 2705 training samples with an accuracy of 46.17%, and Motion Sense has 3472 with an accuracy of 78.79%.
- m = 15, MobiAct has 3263 training samples with an accuracy of 86.32%, UnIMiB-SHAR has 3837 training samples with an accuracy of 46.77%, and Motion Sense has 5280 with an accuracy of 80.25%.
- m = 20, MobiAct has 4345 training samples with an accuracy of 86.96%, UnIMiB-SHAR has 5020 training samples with an accuracy of 46.77%, and Motion Sense has 6952 with an accuracy of 80.41%.

In general, even though MobiAct presents less training samples, it outperforms Motion Sense. UniMiB-SHAR has, in general, a different behaviour in comparison with the other datasets, since the training data size has not a relevant influence in the models performance. For instance, from $m = 10$ to m = 20 the accuracy does not show a significant improvement.

The dependency between the algorithm performance and the dataset size can be more appreciated in Figure 4.3, where the training size for a given m against the accuracy of the PDL models are depicted. In details, on the x-axes the number of the m nearest subject is displayed, while the left y-axes represents the accuracy ± standard deviation. In the same graph, in orange, the barplot represents the frequency distribution of the total number of the samples belonging to the training dataset, with respect to the number of subjects m and the right y-axes represents the total number of samples depending on m. The Figure refers to most to the hybrid model.

These results show that the personalization is effective for the algorithm performances and can improve the algorithm accuracy, even though with less samples. We can state that taking into account the subject similarities plays a relevant role for the PDL performances.
In Figure 4.2 we depicted the similarity matrices of MobiAct, UniMiB-SHAR and Motion Sense split into physical, sensors and the combination between physical and sensor ($\gamma = 1$). We can notice that UniMiB-SHAR presents very low differences between subjects. In other words, subjects in UniMiB-SAHR are very similar to each other. In opposite, subjects in MobiAct and Motion Sense present higher variability. It results that the more the differences between users, the more the personalization is affective even with small samples size.

- *performances across PDL with PML*: In MobiAct and UniMiB-SHAR datasets, PML models overcome PDL strategy in most of the cases.
  In MobiAct dataset only with impersonal model and physical similarity the PDL model outperforms PML. The best performance in MobiAct dataset is achieved using hybrid model with the combination of physical and sensor attributes with an accuracy equal to 90.90%. In average PML achieve 86.46% of accuracy, about 4% more than PDL accuracy.
  In UniMiB-SHAR dataset, PML models achieve better performance than PDL in all of the cases. The best accuracy of 84.47% corresponds to hybrid model with the combination of physical and sensor attributes. In this case, the margin with respect to the corresponding PDL is of 38.10%. In average PML accuracy achieves 71.05%, while PDL only 43.46%, with a margin of 27.59%.
  In Motion Sense dataset shows a completely different behaviour. PDL models performances always outperform PML accuracy. The best models is the hybrid model with the combination of physical and sensor attributes, which reaches the 80.41% of accuracy. The corresponding PML models achieve 77.86% by a margin of 2.55%. In average, PDL models achieve an accuracy of 79.46% by a margin of 3.8% to PML.

  In general, the differences between PDL and PML in MobiAct and Motion Sense are not relevant, while, on the opposite, in the case of UniMiB-SHAR, PML models provide a relevant improvement to the classification performance, with a margin of about 27% from PDL models. Given the similarity matrices in Figure 4.2, it is likely that PML models can better handle with datasets with small differences between subjects. In contrast, PDL models are more effective when subjects difference are higher independently on the sample size.

4.4.4 Conclusions

The goal of this section was to evaluate the performance of personalized machine learning and personalized deep learning models.

Results show that the choice between personalized machine learning and personalized deep learning techniques is not obvious. On one hand, we showed that PML drastically improves the accuracy in the UniMiB-SHAR dataset, where the variability between subject in terms of similarity is low. PML remains the better solution for MobiAct and UniMiB-SHAR datasets. On the other hand, PDL improves the algorithm performances independently on the sample size, as in the case of MobiAct and Motion Sense. It is likely that the personalization of those techniques is more effective when the subjects variability is high.

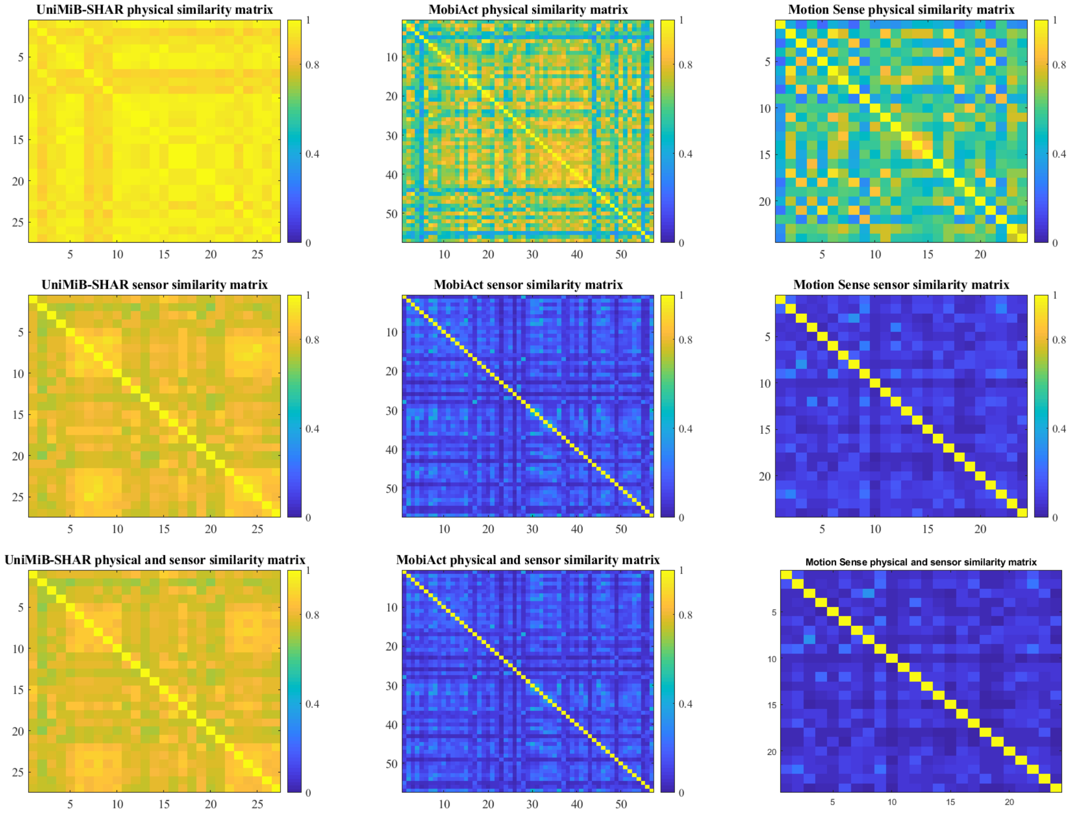


Figure 4.2: Similarity matrices for physical, sensor and their combinations of UniMiB-SHAR, MobiAct and Motion Sense datasets.

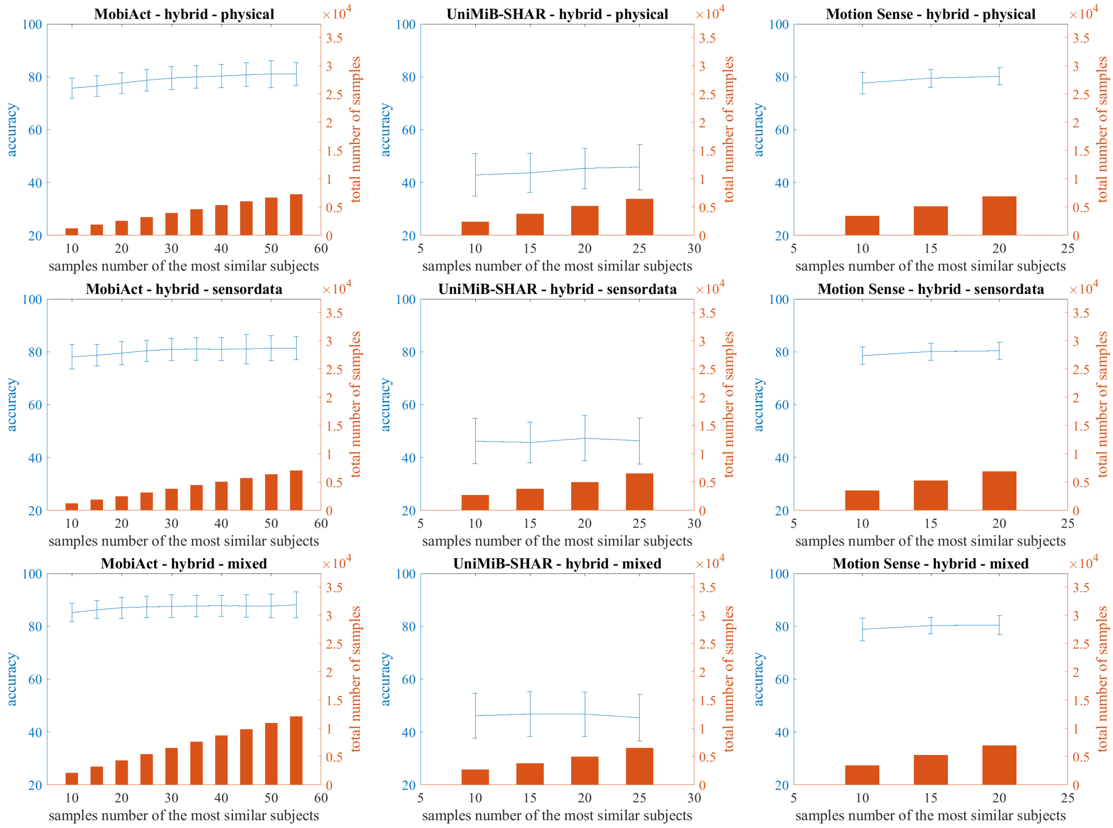


Figure 4.3: hybrid PDL models performances with different training number m of test's nearest subjects (blue line) ± standard deviation and sample frequency distribution (orange bars)

Chapter 5

Conclusions

In this work we compared machine learning and deep learning models based on traditional and personalized approaches. All approaches have been evaluated on the different data splits, namely, subject - independent, and hybrid in combination with the similarity-based personalization, as explained in Chapter 3. In particular, traditional and personalized deep learning methods are based on Convolutional Neural Networks. Traditional and Personalized Machine learning approaches are based on AdaBoost classifier.

Table 5.1 presents an overview of all results achieved in this work. It summarizes the accuracy achieved from personalized deep learning (PDL), personalized machine learning (PML), traditional deep learning (DL), and traditional machine learning (ML) splits into MobiAct, UniMiB-SHAR and Motion Sense datasets. Results are subdivided into data splits, subject - independent and hybrid, and into personalized or traditional. In the last row the overall accuracy average is shown.

DL models outperform the other strategies in the most of the cases. PML overcomes DL only in the case of UniMiB-SHAR dataset with hybrid models. Nevertheless, in UniMiB-SHAR, DL strategies improve the overall accuracy in comparison with ML and PDL methods.

On total average, PDL models achieve an accuracy equal to 68.56%, PML of 77.73%, DL of 79.49%, and ML of 71.63%. DL models improve the performance of at least about 2%. DL models show, in general, better results on MobiAct dataset with an accuracy equal to 92.62% with hybrid model. In the case of subject - independent the 88.92% is achieved. That is an expected behaviour because MobiAct is the largest dataset, which generally improves the classification capability.

On UniMiB-SHAR, the best accuracy is achieved from the PML with hybrid model (84.87%). Nevertheless, in the subject-independent model, DL still achieves the highest accuracy of 58.88%. Accuracy achieved with Motion Sense presents in average the 83.39%, by a margin from 4 and 10% with respect to the other techniques.

Dataset	**Models**	**PDL**	**PML**	**DL**	**ML**
MobiAct	subject - independent - traditional	-	-	88.92	81.29
	subject - independent- physical	86.08	81.62		
	subject - independent - sensor	80.14	83.45		
	subject - independent - physical - sensor	79.68	82.64		
	hybrid - traditional	-	-	92.62	83.73
	hybrid - physical	81.04	89.43		
	hybrid - sensor	81.40	90.76		
	hybrid - physical - sensor	88.17	90.90		
average		82.75	86.46	**90.77**	82.51
UniMiB-SHAR	subject - independent - traditional	-	-	58.88	56.80
	subject - independent- physical	35.42	57.39		
	subject - independent - sensor	42.83	57.00		
	subject - independent - physical - sensor	42.66	56.93		
	hybrid - traditional	-	-	69.72	61.66
	hybrid - physical	45.82	85.44		
	hybrid - sensor	47.26	84.71		
	hybrid - physical - sensor	46.77	84.87		
average		43.46	**71.05**	64.30	59.23
Motion Sense	subject - independent - traditional	-	-	81.03	72.48
	subject - independent - physical	78.02	72.45		
	subject - independent - sensor	78.8	74.03		
	subject - independent - physical - sensor	79.00	73.85		
	hybrid - traditional	-	-	85.75	73.82
	hybrid - physical	80.17	77.76		
	hybrid - sensor	80.38	78.06		
	hybrid - physical - sensor	80.41	77.86		
average		79.46	75.66	**83.39**	73.15
total average		68.56	77.73	**79.49**	71.63

Table 5.1: Experimental Results - accuracy of personalized deep learning (PDL), personalized machine learning (PML).

These results show that DL models are the most preferable in terms of robustness in comparison with PML, PDL, and ML techniques. Indeed, DL based performance outperforms the other method's performances even with different data split and different training datasets. The variability inter and intra subject is overcome by DL. This result allows us to consider DL the method, which achieves the highest generalization capability. The comparison between DL and PDL methods lead us to state that the training dataset size highly influences the algorithm's performance and normally large dataset are preferable. Indeed, the difference between PDL and DL methods is the training dataset's size.

In conclusion, we state that DL algorithms are able to generalize user's differences and show very robust properties in terms of subject's variabilities. Even though the results are based on small scale datasets, DL remain very performant and powerful HAR methods.

Summary of Contributions
This work focused on the improvement of machine learning and deep learning techniques

in terms of generalization capability to face to new unseen user. In the following, the research questions and the related results are summarized.

- Novel classification models based on the personalization of machine learning techniques have been proposed. In particular, the personalization consists in integrating the traditional machine learning algorithms with the metadata of the subjects, such as weighs, height and age, and characteristics relative to the sensor's signals. The reliant research question is "Does Personalized Machine Learning outperform Traditional Machine Learning techniques?" Results demonstrate that the personalization is a valid solution to overcome generalization issues and lead to more performant results, compared to those in the state-of-the-art. In average, the best accuracy have been achieved by the hybrid model with sensor's signals similarity [44].

- Among machine learning and deep learning techniques, it is not clear which methods is more appropriate, mostly in small size-based training datasets. The research question "Does Deep Learning outperform Traditional Machine Learning techniques?" aims at investigating which of those traditional techniques is more suitable in the HAR context. In particular, different feature extraction procedures have been analyzed and compared. Results show that deep learning methods outperform all machine learning configurations and present relevant robustness regarding the choice of the input features. Traditional deep learning techniques remain preferable even trained on small datasets [43, 42].

- Promising results about generalization capability of traditional deep learning techniques lead to the comparison between traditional machine learning and personalized machine learning, namely "Does Deep Learning outperform Personalized Machine Learning? Results show that in average deep learning outperform personalized machine learning techniques. These results highlight the capability of deep learning techniques in generalized over different user's characteristics. A last crucial comparison have been done between personalized deep learning and personalized machine learning as follows [45].

- Promising results in using deep learning methods, stimulated us to experiment novel personalized deep learning models to be compared with the personalized machine learning models above mentioned. Thus, based on the similarity matrix, specific samples have been selected to be part of the training dataset in the deep learning training phase. The related research question is "Does Personalized Deep Learning outperform Personalized and Traditional Machine Learning?". Results show that in some cases the personalized machine learning models outperforms the personalized deep learning ones. The performance of both methods are highly dependent on the subject's similarities within the dataset [45].

All algorithms have been trained and tested on public dataset for guaranteeing the reproducibility of the results.

Bibliography

[1] Zahraa Said Abdallah, Mohamed Medhat Gaber, Bala Srinivasan, and Shonali Krishnaswamy. Adaptive mobile activity recognition system with evolving data streams. *Neurocomputing*, 150:304–317, 2015.

[2] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[3] Bandar Almaslukh, Abdel Monim Artoli, and Jalal Al-Muhtadi. A robust deep learning approach for position-independent smartphone-based human activity recognition. *Sensors*, 18(11):3726, 2018.

[4] Abdulrahman Alruban, Hind Alobaidi, Nathan Clarke, and Fudong Li. Physical activity recognition by utilising smartphone sensor signals. In *8th International Conference on Pattern Recognition Applications and Methods*, pages 342–351. SciTePress, 2019.

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

[6] Ilham Amezzane, Youssef Fakhri, Mohamed El Aroussi, and Mohamed Bakhouya. Towards an efficient implementation of human activity recognition for mobile devices. *EAI Endorsed Transactions on Context-Aware Systems and Applications*, 4(13), 2018.

[7] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN13)*, 2013.

[8] P Antal. Construction of a classifier with prior domain knowledge formalised as bayesian network. In *IECON'98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No. 98CH36200)*, volume 4, pages 2527–2531. IEEE, 1998.

[9] Erik K Antonsson and Robert W Mann. The frequency content of gait. *Journal of biomechanics*, 18(1):39–47, 1985.

[10] Fabio Bagala, Clemens Becker, Angelo Cappello, Lorenzo Chiari, Kamiar Aminian, Jeffrey M Hausdorff, Wiebren Zijlstra, and Jochen Klenk. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PloS one*, 7(5):e37062, 2012.

[11] Oresti Banos, Juan-Manuel Galvez, Miguel Damas, Hector Pomares, and Ignacio Rojas. Window size impact in human activity recognition. *Sensors*, 14(4):6474–6499, 2014.

[12] Aakash Bansal, Abhishek Shukla, Shaurya Rastogi, and Sangeeta Mittal. Micro activity recognition of mobile phone users using inbuilt sensors. In *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 225–230. IEEE, 2018.

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

[14] Stephen D Bay, Dennis Kibler, Michael J Pazzani, and Padhraic Smyth. The uci kdd archive of large data sets for data mining research and experimentation. *ACM SIGKDD explorations newsletter*, 2(2):81–85, 2000.

[15] Akram Bayat, Marc Pomplun, and Duc A Tran. A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science*, 34:450–457, 2014.

[16] Martin Berchtold, Matthias Budde, Hedda R Schmidtke, and Michael Beigl. An extensible modular recognition concept that makes activity recognition practical. In *Annual Conference on Artificial Intelligence (AAAI)*, 2010.

[17] Pratool Bharti, Debraj De, Sriram Chellappan, and Sajal K Das. Human: Complex activity recognition with multi-modal multi-positional body sensing. *IEEE Transactions on Mobile Computing*, 18(4):857–870, 2018.

[18] Valentina Bianchi, Marco Bassoli, Gianfranco Lombardo, Paolo Fornacciari, Monica Mordonini, and Ilaria De Munari. Iot wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment. *IEEE Internet of Things Journal*, 6(5):8553–8562, 2019.

[19] Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277, 2018.

[20] Simone Bianco, Paolo Napoletano, and Raimondo Schettini. Multimodal car driver stress recognition. In *Proceedings of the EAI International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth19)*, 2019.

[21] Christopher M Bishop. *Pattern recognition and machine learning*. Springer-Verlag New York, 2006.

[22] Igor Bisio, Alessandro Delfino, Fabio Lavagetto, and Andrea Sciarrone. Enabling iot for in-home rehabilitation: Accelerometer signals classification methods for activity and movement recognition. *IEEE Internet of Things Journal*, 4(1):135–146, 2016.

[23] Xiao Bo, Alan Huebner, Christian Poellabauer, Megan K O'Brien, Chaithanya Krishna Mummidisetty, and Arun Jayaraman. Evaluation of sensing and processing parameters for human action recognition. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 541–546. IEEE, 2017.

[24] Leo Breiman. 1 random forests–random features. 1999.

[25] Michael Buettner, Richa Prasad, Matthai Philipose, and David Wetherall. Recognizing daily activities with rfid-based sensors. In *Proceedings of the 11th international conference on Ubiquitous computing*, pages 51–60, 2009.

[26] David M Burns and Cari M Whyne. Personalized activity recognition with deep triplet embeddings. *arXiv preprint arXiv:2001.05517*, 2020.

[27] Hilary Buxton. Learning and understanding dynamic scene activity: a review. *Image and vision computing*, 21(1):125–136, 2003.

[28] Nicole A Capela, Edward D Lemaire, and Natalie Baddour. Improving classification of sit, stand, and lie in a smartphone human activity recognition system. In *2015 IEEE International Symposium on Medical Measurements and Applications (MeMeA) Proceedings*, pages 473–478. IEEE, 2015.

[29] Eduardo Casilari, Jose A Santoyo-Ramón, and Jose M Cano-García. Umafall: A multisensor dataset for the research on automatic fall detection. *Procedia Computer Science*, 110:32–39, 2017.

[30] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: overview, challenges and opportunities. *arXiv preprint arXiv:2001.07416*, 2020.

[31] Yufei Chen and Chao Shen. Performance analysis of smartphone-sensor behavior for human activity recognition. *Ieee Access*, 5:3095–3110, 2017.

[32] Jingyuan Cheng, Mathias Sundholm, Bo Zhou, Marco Hirsch, and Paul Lukowicz. Smart-surface: Large scale textile pressure sensors arrays for activity recognition. *Pervasive and Mobile Computing*, 30:97–112, 2016.

[33] Varoth Chotpitayasunondh and Karen M Douglas. How “phubbing” becomes the norm: The antecedents and consequences of snubbing via smartphone. *Computers in Human Behavior*, 63:9–18, 2016.

[34] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[35] Serhan Coşar, Giuseppe Donatiello, Vania Bogorny, Carolina Garate, Luis Otavio Alvares, and François Brémond. Toward abnormal trajectory and event detection in video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):683–695, 2016.

[36] Trevor F Cox and Michael AA Cox. *Multidimensional scaling*. Chapman and hall/CRC, 2000.

[37] Florenc Demrozi, Graziano Pravadelli, Azra Bihorac, and Parisa Rashidi. Human activity recognition using inertial, physiological and environmental sensors: a comprehensive survey. *arXiv preprint arXiv:2004.08821*, 2020.

[38] I Elamvazuthi, Lila Iznita Izhar, Genci Capi, et al. Classification of human daily activities using ensemble methods based on smartphone inertial sensors. *Sensors*, 18(12):4132, 2018.

[39] Anna Ferrari, Daniela Micucci, Mobilio Marco, and Paolo Napoletano. A framework for long-term data collection to support automatic human activity recognition. In *Proceedings of Intelligent Environments: Workshop on Reliable Intelligent Environment (IE 19)*, 2019.

[40] Anna Ferrari, Daniela Micucci, Mobilio Marco, and Paolo Napoletano. Hand-crafted features vs residual networks for human activities recognition using accelerometer. In *Proceedings of the IEEE International Symposium on Consumer Technologies (ISCT)*, 2019.

[41] Anna Ferrari, Daniela Micucci, Mobilio Marco, and Paolo Napoletano. On the homogenization of heterogeneous inertial-based databases for human activity recognition. In *Proceedings of IEEE SERVICES Workshop on Big Data for Public Health Policy Making*, 2019.

[42] Anna Ferrari, Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Hand-crafted features vs residual networks for human activities recognition using accelerometer.

In *2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT)*, pages 153–156. IEEE, 2019.

[43] Anna Ferrari, Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Human activities recognition using accelerometer and gyroscope. In *European Conference on Ambient Intelligence*, pages 357–362. Springer, 2019.

[44] Anna Ferrari, Daniela Micucci, Marco Mobilio, and Paolo Napoletano. On the personalization of classification models for human activity recognition. *IEEE Access*, 8:32066–32079, 2020.

[45] Anna Ferrari, Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Personalization in human activity recognition. *arXiv preprint arXiv:2009.00268*, 2020.

[46] Anna Ferrari, Marco Mobilio, Daniela Micucci, and Paolo Napoletano. On the homogenization of heterogeneous inertial-based databases for human activity recognition. In *2019 IEEE World Congress on Services (SERVICES)*, volume 2642, pages 295–300. IEEE, 2019.

[47] Friedrich Foerster, Manfred Smeja, and Jochen Fahrenberg. Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Computers in human behavior*, 15(5):571–583, 1999.

[48] Nicholas Foubert, Anita M McKee, Rafik A Goubran, and Frank Knoefel. Lying and sitting posture recognition and transition detection using a pressure sensor array. In *2012 IEEE International Symposium on Medical Measurements and Applications Proceedings*, pages 1–6. IEEE, 2012.

[49] Nweke Henry Friday, Mohammed Ali Al-garadi, Ghulam Mujtaba, Uzoma Rita Alo, and Ahmad Waqas. Deep learning fusion conceptual frameworks for complex human activity recognition using mobile and wearable sensors. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–7. IEEE, 2018.

[50] Enrique Garcia-Ceja and Ramon Brena. Building personalized activity recognition models with scarce labeled data based on class similarities. In *International conference on ubiquitous computing and ambient intelligence*, pages 265–276. Springer, 2015.

[51] Enrique Garcia-Ceja and Ramon Brena. Activity recognition using community data to complement small amounts of labeled instances. *Sensors*, 16(6):877, 2016.

[52] Davide Ginelli, Daniela Micucci, Marco Mobilio, and Paolo Napoletano. UniMiB AAL: An Android Sensor Data Acquisition and Labeling Suite. *Applied Sciences*, 8(8), 2018.

[53] Alan Godfrey, Victoria Hetherington, H Shum, Paolo Bonato, NH Lovell, and S Stuart. From a to z: Wearable technology explained. *Maturitas*, 113:40–47, 2018.

[54] Jordi Gonzàlez, Thomas B Moeslund, Liang Wang, et al. Semantic understanding of human behaviors in image sequences: From video-surveillance to video-hermeneutics. *Computer Vision and Image Understanding*, 116(3):305–306, 2012.

[55] Donghai Guan, Tinghuai Ma, Weiwei Yuan, Young-Koo Lee, and AM Jehad Sarkar. Review of sensor-based activity recognition systems. *IETE Technical Review*, 28(5):418–433, 2011.

[56] Niels Haering, Péter L Venetianer, and Alan Lipton. The evolution of video surveillance: an overview. *Machine Vision and Applications*, 19(5-6):279–290, 2008.

[57] Nils Y Hammerla, Shane Halloran, and Thomas Plötz. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv preprint arXiv:1604.08880*, 2016.

[58] Mohammed Mehedi Hassan, Md Zia Uddin, Amr Mohamed, and Ahmad Almogren. A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, 81:307–313, 2018.

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)16)*, pages 770–778, 2016.

[60] Fabio Hernández, Luis F Suárez, Javier Villamizar, and Miguel Altuve. Human activity recognition on smartphones using a bidirectional lstm network. In *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, pages 1–5. IEEE, 2019.

[61] Jin-Hyuk Hong, Julian Ramos, and Anind K Dey. Toward personalized activity recognition systems with a semipopulation approach. *IEEE Transactions on Human-Machine Systems*, 46(1):101–112, 2016.

[62] Duy Tam Gilles Huynh. *Human activity recognition with wearable sensors*. PhD thesis, Technische Universitat, 2008.

[63] Raul Igual, Carlos Medrano, and Inmaculada Plaza. A comparison of public datasets for acceleration-based fall detection. *Medical engineering & physics*, 37(9):870–878, 2015.

[64] Ahmad Jalal, Majid Ali Khan Quaid, and Abdul S Hasan. Wearable sensor-based human behavior understanding and recognition in daily life for smart environments.

In *2018 International Conference on Frontiers of Information Technology (FIT)*, pages 105–110. IEEE, 2018.

[65] Majid Janidarmian, Atena Roshan Fekr, Katarzyna Radecka, and Zeljko Zilic. A comprehensive analysis on wearable acceleration sensors in human activity recognition. *Sensors*, 17(3):529, 2017.

[66] Ian T Jolliffe. Principal component analysis: a beginner's guide. introduction and application. *Weather*, 45(10):375–382, 1990.

[67] Nobuo Kawaguchi, Hodaka Watanabe, Tianhui Yang, Nobuhiro Ogawa, Yohei Iwasaki, Katsuhiko Kaji, Tsutomu Terada, Kazuya Murao, Hisakazu Hada, Sozo Inoue, et al. Hasc2012corpus: Large scale human activity corpus and its application. In *Proceedings of the Second International Workshop of Mobile Sensing: From Smartphones and Wearables to Big Data*, pages 10–14, 2012.

[68] Adil Mehmood Khan, Y-K Lee, Seok-Yong Lee, and T-S Kim. Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *2010 5th international conference on future information technology*, pages 1–6. IEEE, 2010.

[69] Christian Krupitzer, Timo Sztyler, Janick Edinger, Martin Breitbach, Heiner Stuckenschmidt, and Christian Becker. Hips do lie! a position-aware mobile fall detection system. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10. IEEE, 2018.

[70] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[71] Paula Lago and Sozo Inoue. Comparing feature learning methods for human activity recognition: Performance study in new user scenario. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 118–123. IEEE, 2019.

[72] Nicholas D Lane, Ye Xu, Hong Lu, Shaohan Hu, Tanzeem Choudhury, Andrew T Campbell, and Feng Zhao. Enabling large-scale human activity inference on smartphones using community similarity networks (csn). In *Proceedings of the International Conference on Ubiquitous Computing (UbiComp)*, 2011.

[73] Oscar D Lara, Miguel A Labrador, et al. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013.

[74] Oscar D Lara, Alfredo J Pérez, Miguel A Labrador, and José D Posada. Centinela: A human activity recognition system based on acceleration and vital sign data. *Pervasive and mobile computing*, 8(5):717–729, 2012.

[75] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[76] Song-Mi Lee, Sang Min Yoon, and Heeryon Cho. Human activity recognition from accelerometer data using convolutional neural network. In *2017 IEEE International conference on big data and smart computing (BigComp)*, pages 131–134. IEEE, 2017.

[77] Frédéric Li, Kimiaki Shirahama, Muhammad Adeel Nisar, Lukas Köping, and Marcin Grzegorzek. Comparison of feature learning methods for human activity recognition using wearable sensors. *Sensors*, 18(2):679, 2018.

[78] Xinyu Li, Yuan He, and Xiaojun Jing. A survey of deep learning-based human activity recognition in radar. *Remote Sensing*, 11(9):1068, 2019.

[79] Yingzi Lin and WJ Zhang. Towards a novel interface design framework: function–behavior–state paradigm. *International journal of human-computer studies*, 61(3):259–297, 2004.

[80] Huan Liu and Hiroshi Motoda. *Feature extraction, construction and selection: A data mining perspective*, volume 453. Springer Science and Business Media, 1998.

[81] Juzheng Liu, Jing Chen, Hanjun Jiang, Wen Jia, Qingliang Lin, and Zhihua Wang. Activity recognition in wearable ecg monitoring aided by accelerometer data. In *2018 IEEE international symposium on circuits and systems (ISCAS)*, pages 1–4. IEEE, 2018.

[82] Jeffrey W Lockhart and Gary M Weiss. The benefits of personalized smartphone-based activity recognition models. In *Proceedings of the 2014 SIAM international conference on data mining*, pages 614–622. SIAM, 2014.

[83] Jeffrey W Lockhart and Gary M Weiss. Limitations with activity recognition methodology & data sets. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 747–756, 2014.

[84] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D Lane, Tanzeem Choudhury, and Andrew T Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the ACM conference on embedded networked sensor systems (SenSys)*, 2010.

[85] Amira Ben Mabrouk and Ezzeddine Zagrouba. Abnormal behavior recognition for intelligent video surveillance systems: A review. *Expert Systems with Applications*, 91:480–491, 2018.

[86] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Protecting sensory data against sensitive inferences. In *Proceedings of the Workshop on Privacy by Design in Distributed Systems (W-P2DS18)*, 2018.

[87] Carlos Medrano, Raul Igual, Inmaculada Plaza, and Manuel Castro. Detecting falls as novelties in acceleration patterns acquired with smartphones. *PloS one*, 9(4):e94811, 2014.

[88] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10):1101, 2017.

[89] Daniela Micucci, Marco Mobilio, Paolo Napoletano, and Francesco Tisato. Falls as anomalies? an experimental evaluation using smartphone accelerometer data. *Journal of Ambient Intelligence and Humanized Computing*, 8(1):87–99, 2017.

[90] Martin Milenkoski, Kire Trivodaliev, Slobodan Kalajdziski, Mile Jovanov, and Biljana Risteska Stojkoska. Real time human activity recognition on smartphones using lstm networks. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1126–1131. IEEE, 2018.

[91] Kartik Muralidharan, Azeem Javed Khan, Archan Misra, Rajesh Krishna Balan, and Sharad Agarwal. Barometric phone sensors: More hype than hope! In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, pages 1–6, 2014.

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

[93] HD Nguyen, Kim Phuc Tran, X Zeng, Ludovic Koehl, and Guillaume Tartare. Wearable sensor data based human activity recognition using machine learning: A new approach. *arXiv preprint arXiv:1905.03809*, 2019.

[94] Luís ML Oliveira and Joel JPC Rodrigues. Wireless sensor networks: A survey on environmental monitoring. *JCM*, 6(2):143–151, 2011.

[95] Francisco Javier Ordonez, Gwenn Englebienne, Paula De Toledo, Tim Van Kasteren, Araceli Sanchis, and Ben Kröse. In-home activity recognition: Bayesian inference for hidden markov models. *IEEE Pervasive Computing*, 13(3):67–75, 2014.

[96] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, 150:109–125, 2016.

[97] Leslie A Perlow. *Sleeping with your smartphone: How to break the 24/7 habit and change the way you work.* Harvard Business Press, 2012.

[98] Thomas Plötz, Nils Y Hammerla, and Patrick L Olivier. Feature learning for activity recognition in ubiquitous computing. In *Twenty-second international joint conference on artificial intelligence*, 2011.

[99] Sandeep Kumar Polu. Human activity recognition on smartphones using machine learning algorithms. *International Journal for Innovative Research in Science & Technology*, 5(6):31–37, 2018.

[100] Jun Qi, Po Yang, Atif Waraich, Zhikun Deng, Youbing Zhao, and Yun Yang. Examining sensor-based physical activity recognition and monitoring for healthcare using internet of things: A systematic review. *Journal of biomedical informatics*, 87:138–153, 2018.

[101] Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *Proceedings of the Conference on Innovative applications of Artificial Intelligence (IAAI)*, 2005.

[102] Attila Reiss and Didier Stricker. Personalized mobile physical activity recognition. In *Proceeding of the IEEE International Symposium on Wearable Computers (ISWC)*, 2013.

[103] Lior Rokach and Oded Z Maimon. *Data mining with decision trees: theory and applications*, volume 69. World scientific, 2008.

[104] Seyed Ali Rokni, Marjan Nourollahi, and Hassan Ghasemzadeh. Personalized human activity recognition using convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[105] Charissa Ann Ronao and Sung-Bae Cho. Human activity recognition using smartphone sensors with two-stage continuous hidden markov models. In *2014 10th International Conference on Natural Computation (ICNC)*, pages 681–686. IEEE, 2014.

[106] Charissa Ann Ronao and Sung-Bae Cho. Deep convolutional neural networks for human activity recognition with smartphone sensors. In *International Conference on Neural Information Processing*, pages 46–53. Springer, 2015.

[107] Charissa Ann Ronao and Sung-Bae Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, 59:235–244, 2016.

[108] Sadiq Sani, Stewart Massie, Nirmalie Wiratunga, and Kay Cooper. Learning deep and shallow features for human activity recognition. In *International Conference on Knowledge Science, Engineering and Management*, pages 469–482. Springer, 2017.

[109] Sarbagya Ratna Shakya, Chaoyang Zhang, and Zhaoxian Zhou. Comparative study of machine learning and deep learning architecture for human activity recognition using accelerometer data. *Int. J. Mach. Learn. Comput*, 8:577–582, 2018.

[110] Chao Shen, Yufei Chen, and Gengshan Yang. On motion-sensor behavior analysis for human-activity recognition via smartphones. In *2016 Ieee International Conference on Identity, Security and Behavior Analysis (Isba)*, pages 1–6. IEEE, 2016.

[111] Kimiaki Shirahama and Marcin Grzegorzek. On the generality of codebook approach for sensor-based human activity recognition. *Electronics*, 6(2):44, 2017.

[112] Muhammad Shoaib, Stephan Bosch, Ozlem Incel, Hans Scholten, and Paul Havinga. A survey of online activity recognition using mobile phones. *Sensors*, 15(1):2059–2085, 2015.

[113] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, 14(6):10146–10176, 2014.

[114] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. Complex human activity recognition using smartphone and wrist-worn motion sensors. *Sensors*, 16(4):426, 2016.

[115] Muhammad Shoaib, Hans Scholten, and Paul JM Havinga. Towards physical activity recognition using smartphone sensors. In *2013 IEEE 10th international conference on ubiquitous intelligence and computing and 2013 IEEE 10th international conference on autonomic and trusted computing*, pages 80–87. IEEE, 2013.

[116] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5734–5743, 2017.

[117] Pekka Siirtola, Heli Koskimäki, and Juha Röning. Openhar: A matlab toolbox for easy access to publicly open human activity data sets. In *Proceedings of the ACM International Joint Conference and International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers (UbiComp18)*, 2018.

[118] Pekka Siirtola, Heli Koskimäki, and Juha Röning. From user-independent to personal human activity recognition models exploiting the sensors of a smartphone. *arXiv preprint arXiv:1905.12285*, 2019.

[119] Pekka Siirtola, Heli Koskimäki, and Juha Röning. Personalizing human activity recognition models using incremental learning. *arXiv preprint arXiv:1905.12628*, 2019.

[120] Pekka Siirtola and Juha Röning. Recognizing human activities user-independently on smartphones based on accelerometer data. *IJIMAI*, 1(5):38–45, 2012.

[121] Pekka Siirtola and Juha Röning. Incremental learning to personalize human activity recognition models: The importance of human ai collaboration. *Sensors*, 19(23):5151, 2019.

[122] Sithara P Sreenilayam, Inam Ul Ahad, Valeria Nicolosi, Victor Acinas Garzon, and Dermot Brabazon. Advanced materials of printed wearables for physiological parameter monitoring. *Materials Today*, 32:147–177, 2020.

[123] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjaergaard, Anind Dey, Tobias Sonne, and Mads Moeller Jensen. Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 127–140, 2015.

[124] Xing Su, Hanghang Tong, and Ping Ji. Accelerometer-based activity recognition on smartphone. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 2021–2023, 2014.

[125] Xing Su, Hanghang Tong, and Ping Ji. Activity recognition with smartphone sensors. *Tsinghua science and technology*, 19(3):235–249, 2014.

[126] Jozsef Suto, Stefan Oniga, Claudiu Lung, and Ioan Orha. Comparison of offline and real-time human activity recognition results using machine learning techniques. *Neural computing and applications*, pages 1–14, 2018.

[127] Timo Sztyler and Heiner Stuckenschmidt. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9. IEEE, 2016.

[128] Timo Sztyler and Heiner Stuckenschmidt. Online personalization of cross-subjects based activity recognition models on wearable devices. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2017.

[129] Timo Sztyler, Heiner Stuckenschmidt, and Wolfgang Petrich. Position-aware activity recognition with wearable devices. *Pervasive and mobile computing*, 38:281–295, 2017.

[130] Emmanuel Munguia Tapia, Stephen S Intille, William Haskell, Kent Larson, Julie Wright, Abby King, and Robert Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *Proceeding of the IEEE International Symposium on Wearable Computers (ISWC)*, 2007.

[131] Md Uddin, Weria Khaksar, Jim Torresen, et al. Ambient sensors for elderly care and independent living: A survey. *Sensors*, 18(7):2027, 2018.

[132] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE Pervasive Computing*, 16(4):62–74, 2017.

[133] Sebastián R Vanrell, Diego H Milone, and H Leonardo Rufiner. Assessment of homomorphic analysis for human activity recognition from acceleration signals. *IEEE journal of biomedical and health informatics*, 22(4):1001–1010, 2018.

[134] Amari Vaughn, Paul Biocco, Yang Liu, and Mohd Anwar. Activity detection and analysis using smartphone sensors. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 102–107. IEEE, 2018.

[135] George Vavoulas, Charikleia Chatzaki, Thodoris Malliotakis, Matthew Pediaditis, and Manolis Tsiknakis. The mobiact dataset: Recognition of activities of daily living using smartphones. In *Proceedings of Information and Communication Technologies for Ageing Well and e-Health (ICT4AgeingWell16)*, 2016.

[136] Quang Viet Vo, Minh Thang Hoang, and Deokjai Choi. Personalization in mobile activity recognition system using k-medoids clustering algorithm. *International Journal of Distributed Sensor Networks*, 9(7):315841, 2013.

[137] Jin Wang, Ping Liu, Mary FH She, Saeid Nahavandi, and Abbas Kouzani. Bag-of-words representation for biomedical time series classification. *Biomedical Signal Processing and Control*, 8(6):634–644, 2013.

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

[139] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Device-free human activity recognition using commercial wifi devices. *IEEE Journal on Selected Areas in Communications*, 35(5):1118–1131, 2017.

[140] Gary M Weiss and Jeffrey W Lockhart. The impact of personalization on smartphone-based activity recognition. In *Proceedings of the AAAI Workshop on Activity Context Representation: Techniques and Languages*, 2012.

[141] Ian H Witten, Eibe Frank, and Mark A Hall. Practical machine learning tools and techniques. *Morgan Kaufmann*, page 578, 2005.

[142] C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth Hayworth, Gary Huang, Kazunori Shinomiya, Jeremy Maitin-Shepard, David Ackerman, Stuart Berg, et al. A connectome of the adult drosophila central brain. *BioRxiv*, 2020.

[143] Wenchao Xu, Yuxin Pang, Yanqin Yang, and Yanbo Liu. Human activity recognition based on convolutional neural network. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 165–170. IEEE, 2018.

[144] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 15)*, 2015.

[145] Tao Yu, Jianxin Chen, Na Yan, and Xipeng Liu. A multi-layer parallel lstm network for human activity recognition with smartphone sensors. In *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2018.

[146] Tong Yu, Yong Zhuang, Ole J Mengshoel, and Osman Yagan. Hybridizing personal and impersonal machine learning models for activity recognition on mobile devices. In *Proceedings of the EAI International Conference on Mobile Computing, Applications and Services (MobiCASE)*, 2016.

[147] Shugang Zhang, Zhiqiang Wei, Jie Nie, Lei Huang, Shuang Wang, and Zhen Li. A review on human activity recognition using vision-based method. *Journal of healthcare engineering*, 2017, 2017.

[148] WJ Zhang, Guosheng Yang, Yingzi Lin, Chunli Ji, and Madan M Gupta. On definition of deep learning. In *2018 World Automation Congress (WAC)*, pages 1–5. IEEE, 2018.

[149] Ran Zhu, Zhuoling Xiao, Ying Li, Mingkun Yang, Yawen Tan, Liang Zhou, Shuisheng Lin, and Hongkai Wen. Efficient human activity recognition solving the confusing activities via deep ensemble learning. *IEEE Access*, 7:75490–75499, 2019.

[150] Muhammad Zia ur Rehman, Asim Waris, Syed Omer Gilani, Mads Jochumsen, Imran Khan Niazi, Mohsin Jamil, Dario Farina, and Ernest Nlandu Kamavuako. Multiday emg-based classification of hand motions with deep learning techniques. *Sensors*, 18(8):2497, 2018.

[151] Andrea Zunino, Jacopo Cavazza, and Vittorio Murino. Revisiting human action recognition: Personalization vs. generalization. In *International Conference on Image Analysis and Processing*, pages 469–480. Springer, 2017.