UNIVERSITÀ DEGLI STUDI DI MILANO

PhD School in
Computer Science

Computer Science Department
"Giovanni Degli Antoni"

PhD in
Computer Science
XXXV Cycle

Collaborative Approaches for Sensor-Based Human Activity Recognition in Data Scarcity Scenarios

INF/01

PhD candidate:
Riccardo PRESOTTO

Advisor:
Prof. Claudio BETTINI

Co-Advisor:
Dr. Gabriele CIVITARESE

School Director:
Prof. Roberto SASSI

Academic Year 2021/2022

Abstract

One of the most important goals of Human Activity Recognition (HAR) is to automatically obtain information on the behaviors of the users to proactively assist them with their tasks. In the literature, the majority of physical activity recognition approaches rely on fully-supervised techniques to collaboratively train a recognition model over the data collected from a large number of users. However, these solutions usually suffer from numerous issues like scalability, privacy, poor personalization, and scarcity of labeled training data. In this thesis, we will focus on analyzing in deep those problems, with the scope of proposing novel methodologies to tackle them. First of all, we consider the labeled data scarcity issue. Indeed, obtaining human-annotated activity examples is costly, intrusive, time-consuming, and hence unpractical on a large scale. Semi-supervised approaches have been suggested to reduce the size of the training set required to initialize the model, but their effectiveness revealed not satisfactory for those activities that involve similar body movements (e.g., standing and taking the elevator). In order to mitigate this problem, we propose a novel hybrid semi-supervised and knowledge-based framework that uses the context that surrounds users (e.g. semantic location, speed, weather) to enable a machine learning model trained with a limited number of labeled data to classify a wide set of context-dependent activities. Then, we consider the scalability and privacy issues that arise in collaboratively training a recognition model with the data coming from a large number of different users. Federated Learning (FL) showed to be a promising paradigm to address these problems. However, most of the FL-based solutions for HAR proposed in the literature assume that users can always obtain labeled data to train the recognition model, hence inheriting the limitation related to human annotation that we mentioned before. Moreover, generating a single global

model for all the users may not be as effective as expected. Indeed, different subjects could perform activities in different ways depending on their physical traits and habits. In order to tackle these problems, we introduce innovative hybrid semi-supervised and FL-based solutions that enable personalized, privacy-aware, and scalable activity recognition. In conclusion, we analyze the possible information leakage of FL for HAR, with the aim of obtaining hints to guide the future development of specific privacy-preserving techniques.

Author's Publications

This thesis is based on the following publications, which have been written during my three years of PhD.

Journals

- Riccardo Presotto, Gabriele Civitarese, Claudio Bettini, *"Federated Clustering and Semi-Supervised Learning: A New Partnership for Personalized Human Activity Recognition"*. Pervasive and Mobile Computing, Elsevier, 2022. (DOI: 10.1016/j.pmcj.2022.101726).

- Riccardo Presotto, Gabriele Civitarese, Claudio Bettini, *"Semi-Supervised and Personalized Federated Activity Recognition Based on Active Learning and Label Propagation"*. Personal and Ubiquitous Computing, Springer, 2022. (DOI: 10.1007/s00779-022-01688-8).

- Claudio Bettini, Gabriele Civitarese, Davide Giancane, Riccardo Presotto, *"ProCAVIAR: Hybrid Data-Driven and Probabilistic Knowledge-Based Activity Recognition"*. IEEE Access, IEEE, 2020. (DOI: 10.1109/ACCESS.2020.3015091).

- Claudio Bettini, Gabriele Civitarese, Riccardo Presotto, *"CAVIAR: Context-driven Active and Incremental Activity Recognition"*. Knowledge-Based Systems, Elsevier, 2020. (DOI: 10.1016/j.knosys.2020.105816).

International conferences

- Riccardo Presotto, Gabriele Civitarese, Claudio Bettini, *"FedCLAR: Federated Clustering for Personalized Sensor-Based Human Activity Recognition"*. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2022.

- Luca Arrotta, Claudio Bettini, Gabriele Civitarese, Riccardo Presotto, *"Context-Aware Data Association for Multi-Inhabitant Sensor-Based Activity Recognition"*. In Proceedings. of the 21st International Conference on Mobile Data Management (MDM), IEEE Computer Society, 2020.

International workshops

- Riccardo Presotto, Gabriele Civitarese, Claudio Bettini. *"Preliminary Results on Sensitive Data Leakage in Federated Human Activity Recognition"*. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications Workshops, 2022.

- Riccardo Presotto. *"Semi-supervised methodologies to tackle the annotated data scarcity problem in the field of HAR"*. 2021 22nd IEEE International Conference on Mobile Data Management (MDM). IEEE, 2021.

- Gabriele Civitarese, Riccardo Presotto, Claudio Bettini. "*Hybrid Data-Driven and Context-Aware Activity Recognition with Mobile Devices*". Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers (UbiComp/ISWC '19 Adjunct), 2019.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation And Problem Description

The rapid evolution of sensor technology and mobile computing in the last decades opened the way to a new generation of intelligent context-aware services able to automatically detect our daily activities [1]. Overall, the main goal of Human Activity Recognition (HAR) is to monitor the behaviors of the users through the analysis of observations obtained from users themselves and their environments of living, with the aim of developing solutions capable of dynamically adapting their functionalities to people's behavior. The contributions in the literature include the development of different AI-based approaches that exploit the data collected from various sources in order to enable smart services like care-giving, home rehabilitation, as well as human well-being and safety [2]. Overall, the typology of activities commonly considered by researchers ranges from Activities of Daily Living (ADLs) like *cooking*, *eating*, *taking medicine*, etc., to physical activities like *walking*, *taking the stairs*, *running*, etc. Numerous research studies have been carried out and many more are in progress in this application domain. In the last two decades, video-based and sensor-based are the two categories of HAR on which most of the works in the literature have been focused [3]. Video-based HAR enables classifying both ADLs and physical activities by processing video or images that contain human actions and motions. However, although the promising results obtained by video-based HAR systems, the privacy and intru-

siveness issues related to continuously monitoring the user by cameras, lead the research community to recently polarise on sensor-based HAR [4].

Considering sensor-based HAR, the adopted sensors equipment varies depending on the category of the activities to classify [5, 6]. ADLs require sophisticated settings (e.g., smart homes) that enable monitoring of both the physical movements of the users and their interaction with the living environment. Differently, physical activities can be classified by processing only the sensor data collected from wearable devices (e.g., smartphones and smartwatches). Indeed, the new generation of wearable devices, embed several sensors such as inertial (e.g., accelerometer, gyroscope) and optical (e.g., heart rate sensor) which enable to seamlessly and unobtrusively obtaining information regarding the physical movements of the users and their health condition [7].

In this thesis, we will focus on physical activity recognition based on the data collected from wearable devices. The fundamental steps adopted to process these data are: pre-processing, segmentation, features extraction, and classification. The pre-processing step usually consists of filtering the sensor data in order to reduce noise. Then, the segmentation process enables the partitioning of the filtered sensor data stream into segments of a specific length. After the segmentation step, the features extraction procedure allows extracting from each segment the most relevant features. In particular, this last step can be automated (i.e., by adopting specific types of Deep Neural Networks (DNN)[8]) or handcrafted [9]. Concerning the classification step, most of the state-of-the-art sensor-based activity recognition systems rely on fully supervised collaborative machine-learning techniques [10, 1]. These approaches mostly involve centralizing a large number of labeled sensor data collected by different subjects into a single cloud server, where a global machine-learning model is trained. Despite the promising results of these approaches, they suffer from various challenges related to scalability, privacy, and personalization. From a privacy perspective, activity data are sensitive since they can reveal users' personal habits and health conditions [11]. Furthermore, the activity recognition process often involves private information, such as the semantic position of the user, which is not meant to be shared or made public. Thus, transferring such data to a third-party cloud server may expose users to

many privacy threats. Considering scalability aspects, fully supervised collaborative machine-learning methods may also pose issues related to communication latency and computational costs. Moreover, we have to consider that different users likely perform activities in very different ways depending on their physical traits. For this reason, it emerges the need for a personalized activity recognition model for each user. In order to achieve this goal, each user would have to collect plenty of labeled examples to train her personal classifier. However, the annotation of activity data is costly, time-consuming, intrusive, and hence often unfeasible on a large scale [12]. Lastly, it is also important to consider that most of the works in the literature exploit only inertial sensor data to classify physical activities, and thus poorly perform in discriminating those activities characterized by similar body movements (e.g., standing and taking the elevator). This issue is even emphasized considering that, as we mentioned above, a single user may struggle to collect the number of annotated examples needed to adequately train a personalized recognition model.

Given all of these possible limitations, in Figure 1.1 we present our vision of an ideal collaborative learning approach for HAR. First of all, we believe that including context data (e.g., semantic location, speed, height variation) along with inertial sensor data, would be important to improve the classification rate of activities characterized by similar body movements. Another important aspect to consider is the minimization of the number of annotated samples that each user has to collect to build a personalized recognition model. Moreover, in our vision, we think that to reduce privacy and scalability issues, each user should share only a small portion of non-sensitive information with a cloud server. On the server side, this non-sensitive information has to be manipulated to generate a specialized recognition model for each group of similar users (e.g., users with comparable physical traits and habits). Lastly, the users should also have the possibility to further personalize the received model over their very distinctive way of executing activities.

In the following of this thesis, we will investigate in deep the limitations related to the most common collaborative HAR approaches. Then, we propose

novel methodologies to tackle them, with the aim of getting as close as possible to our vision of an ideal collaborative learning system for sensor-based HAR.

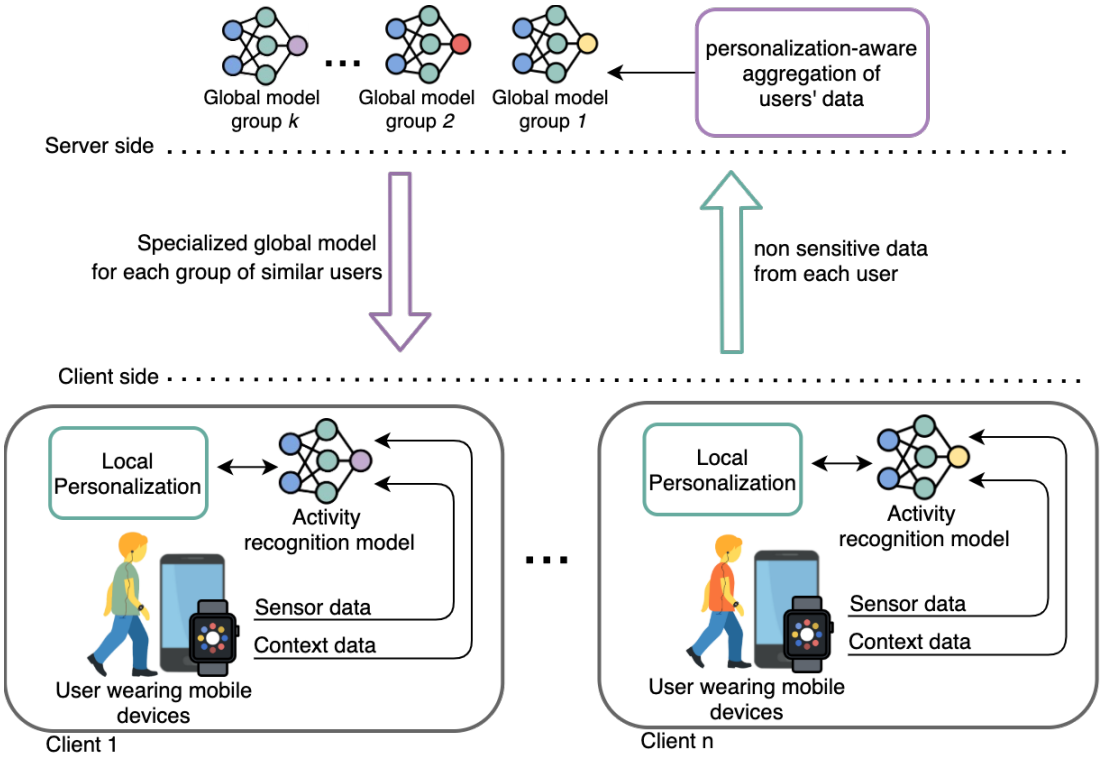


Figure 1.1: Representation of our vision for collaborative approaches for HAR

First of all, focus on the labeled data scarcity problem. Indeed, as we previously introduced, collecting an adequate number of labeled data to train a personalized activity recognition model is a real challenge. For instance, data annotation can be performed directly by each user while performing activities (self-annotation), but this approach is very obtrusive and error-prone. Alternatively, external observers can annotate the activity execution of a subject (in real-time or by semi-automatic video annotation). However, this annotation technique is particularly time-consuming and privacy-intrusive. Semi-supervised and incremental approaches revealed a valuable solution to mitigate this problem [13]. These methods only require a small number of labeled data to initialize the recognition model, while techniques like co-learning, self-learning, or active learning are used to annotate the inertial sensor data stream, and incrementally

train the model [14, 15, 16]. However, due to the small set of labeled data used for model initialization, semi-supervised approaches may struggle in the early stages to discriminate activities that involve similar body movements (e.g., walking and taking the stairs). In order to increase the classification rate of those activities, in the literature, has been proposed to use context data (e.g., semantic location, weather, speed, altitude changes) as additional features in the machine learning process [17, 18]. Anyway, given semi-supervised learning settings, it is not realistic to acquire a comprehensive pre-training dataset that includes the large number of possible context conditions in which activities can be performed. Moreover, since context variables may be high and dynamic, the resulting machine-learning model would be extremely complex.

Along with the annotated data scarcity problem, it is important to consider the scalability and privacy issues that arise in centralizing the data collected by a large number of subjects into a single cloud server, where the global activity recognition model is trained.

Indeed, the data collected from mobile devices and wearable sensors are highly personal and can reveal private and sensitive information about users such as their health status, location, and daily routines [11]. Moreover, as the number of users increases, storing and processing all the collected data on a single machine becomes challenging, making scalability a concern. Further, the computational effort required to train the global model with data coming from numerous users grows significantly with their number.

However, collecting a large number of data from multiple users in a privacy-preserving way is crucial for any effective and privacy-aware collaborative HAR approach. Federated Learning (FL) framework, recently proposed in the literature [19], presents a promising solution to these challenges.

In the FL paradigm, the global model training task is distributed among a large number of nodes. Each node uses its annotated data to train its local instance of the global recognition model. The resulting model parameters of each participating node are then sent to a server, which aggregates them to update the global model. Finally, the updated global model's parameters are shared with the participating nodes.

By sharing the locally learned model parameters instead of data, FL mitigates the scalability and privacy issues of large-scale scenarios. Therefore, FL can be an effective and privacy-aware collaborative approach, providing a valuable solution to the fundamental requirements of collecting and processing vast amounts of data from numerous users while keeping their personal information private. For those reasons, FL attracted attention from the pervasive computing community, including HAR [20, 21, 22].

Despite the potential of FL in HAR scenarios, there are still some limitations. First of all, in the literature most of the FL-based approaches for HAR assume that labeled datasets are available for each client, thus inheriting the data annotation problems that we previously discussed [23]. Then, in FL the server generates a global model with the purpose of generalizing over a large number of different subjects. However, diverse users may perform activities in very different ways and by following diverse routines depending on various factors like habits, physical characteristics, age etc. Accordingly, the data coming from different users is non-independently and identically distributed (non-IID). A trade-off between generalization and personalization should be considered by FL methods to build accurate HAR models [24]. In the literature, the earliest approaches proposed to tackle the non-IID problem for FL HAR rely on transfer learning [23]. Here, the global recognition model is fine-tuned by each participating user by exploiting its locally collected data. However, by depending only on transfer learning and a single global model, it is challenging to balance personalization and generalization, especially considering large-scale scenarios [25].

Lastly, even though FL avoids the release of labeled sensor data, recent studies showed that the model's parameters received and manipulated by the cloud server may still reveal some information about users who generate them [26]. Indeed, deep learning models' parameters could implicitly memorize specific information about the data used to train the model [27, 28]. This problem is particularly relevant in the HAR domain where the training examples may expose private and sensitive information of the users. However, to the best of our knowledge, the potential privacy threats of federated HAR models have not been studied in deep yet.

1.2 Research Contributions

In this section, every research contribution of the thesis is introduced. It is important to note that these contributions have been achieved in collaboration with my research group: the EveryWare Lab [1], at the University of Milan (Italy).

1.2.1 Hybrid Semi-Supervised Learning And Context-Aware Reasoning Framework For Collaborative HAR

As we previously mentioned, state-of-the-art approaches for sensor-based HAR mostly rely on fully-supervised learning strategies to collaboratively train the recognition model. While the literature on this topic is quite mature, existing methodologies do not consider how it is challenging for the users to collect the large number of labeled data required to train the recognition model. Semi-supervised learning approaches may be a valuable solution to mitigate this problem. However, they showed not very effective in classifying activities characterized by similar body movements. Including context data as additional features to train the machine learning model could help to discriminate those activities. Nevertheless, as semi-supervised approaches rely on a small number of examples to initialize the recognition model, collecting a training set including activity data performed in every possible context condition is even more challenging.

In order to tackle these problems, in Chapter 3 we propose a novel approach that combines semi-supervised learning, and knowledge-based reasoning [29, 30]. Precisely, an incremental machine learning classifier is in charge of inferring from the inertial sensors data of each user the probability distribution over the possible activities. Meanwhile, the context data are processed separately by a specific knowledge-based reasoning engine that refines the probability distribution considering context data. Finally, the context-refined predictions are used as newly labeled samples to collaboratively update the classifier by following an active

[1] http://everywarelab.di.unimi.it/

learning-based approach. In particular, we developed and experimentally evaluated two different ontologies to perform the context-based refinement. The first one is a deterministic ontology that enables excluding from the statistical prediction vector the activities which are highly unlikely considering context data [29]. The latter consists of a probabilistic ontology that allows refining the statistical prediction vector by considering the intrinsic uncertainty that characterizes the relationships among activities and context data[30]. The obtained results show that the proposed approach enables improving the recognition rate with respect to state-of-the-art semi-supervised approaches for HAR, while using a very limited number of annotated samples.

Chapter 3 is based on the following publications:

- Claudio Bettini, Gabriele Civitarese, Davide Giancane, Riccardo Presotto, *"ProCAVIAR: Hybrid Data-Driven and Probabilistic Knowledge-Based Activity Recognition"*. IEEE Access, IEEE, 2020. (DOI: 10.1109/ACCESS.2020.3015091).

- Claudio Bettini, Gabriele Civitarese, Riccardo Presotto, *"CAVIAR: Context-driven Active and Incremental Activity Recognition"*. Knowledge-Based Systems, Elsevier, 2020. (DOI: 10.1016/j.knosys.2020.105816).

- Gabriele Civitarese, Riccardo Presotto, Claudio Bettini. "*Hybrid Data-Driven and Context-Aware Activity Recognition with Mobile Devices*". Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and the 2019 International Symposium on Wearable Computers (UbiComp/ISWC '19 Adjunct), 2019.

My Contributions:

- Collaboration in methodology design.
- Collaboration in designing a part of the proposed ontologies.
- System implementation (except ontological reasoning).
- Collaboration in the design of the evaluation methods.
- Experiments execution.
- Collaboration in results analysis and interpretation.

1.2.2 FL-Based Approach To Reduce The Data Scarcity Problem Of HAR

The semi-supervised learning and context-aware reasoning approach proposed in Chapter 3 represents a promising solution to mitigate the labeled data scarcity problem of HAR. However, like most of the state-of-the-art collaborative methods for HAR, the proposed approach involves centralizing the data collected by users on a single machine in order to train a global activity recognition model. Accordingly, scalability and privacy limitations may arise when the process involves a large number of subjects. Federated Learning (FL) is one of the most interesting paradigms to address these problems. Nevertheless, the FL-based approaches for HAR that have been proposed in the literature assume that participating users can always obtain labeled datasets to train their local models.

In Chapter 4, we propose FedAR: a novel approach for HAR that combines semi-supervised and federated learning to take advantage of the strengths of both approaches. FedAR integrates active learning and label propagation to semi-automatically annotate the local streams of unlabeled sensor data, while it relies on FL to build a global activity model in a scalable and privacy-aware fashion. FedAR also includes a transfer learning-inspired strategy to personalize the global model for each user.

We evaluated FedAR on two public datasets, showing that our novel methodology allows achieving a very high recognition rate with a very limited number of annotated training examples, hence leading to an effective, privacy-aware, and scalable solution to tackle the labeled data scarcity problem of HAR.

Chapter 4 is based on the following publication:

- Riccardo Presotto, Gabriele Civitarese, Claudio Bettini, *"Semi-Supervised and Personalized Federated Activity Recognition Based on Active Learning and Label Propagation"*. Personal and Ubiquitous Computing, Springer, 2022.

My Contributions:

- Collaboration in concept and methodology design.
- System implementation.
- Design of the evaluation method.
- Experiments execution.
- Collaboration in results analysis and interpretation.

1.2.3 Tackling The Non-IID Issue Typical Of FL Approaches For HAR

In various domains, the FL paradigm enabled achieving really good results in a scalable and privacy-preserving way. However, considering the specific HAR domain emerged some issues. Indeed, different subjects may perform the same activities in various ways depending on their physical traits, age, habits, etc. The activity data is hence non-independently and identically distributed (non-IID) among the participants. Therefore, the generation of a single global model for all the users may lead to unsatisfactory performances in terms of classification accuracy. The model personalization strategy inspired by transfer learning

that we propose in Chapter 4 partially mitigates this issue. Anyway, personalizing a single global model may not be sufficiently accurate for a large number of users.

In Chapter 5, we introduce SS-FedCLAR, a novel semi-supervised federated learning approach for personalized HAR based on hierarchical clustering. Precisely, SS-FedCLAR relies on an innovative framework that enables grouping the users based on the server-side similarity computation, using only a portion of the model weights shared by each participant. Given the similarity of the local model updates, the cloud server of SS-FedCLAR derives groups of users that exhibit similar ways of performing activities. For each group, SS-FedCLAR generates a specialized global model in order to minimize the non-IID problem. Moreover, SS-FedCLAR takes into account the annotated data scarcity problem: each client uses a combination of active learning and label propagation to provide pseudo labels to a large amount of unlabeled data, which is then used to collaboratively train the Federated Clustering model.

We evaluated SS-FedCLAR on two well-known public datasets. Our results show that it mitigates the non-IID problem and the data scarcity issue at the same time. Indeed, SS-FedCLAR reaches recognition rates that are very close to fully-supervised methods and it outperforms state-of-the-art semi-supervised FL-based HAR approaches.

Chapter 5 is based on the following publications:

- Riccardo Presotto, Gabriele Civitarese, Claudio Bettini, *"Federated Clustering and Semi-Supervised Learning: A New Partnership for Personalized Human Activity Recognition"*. Pervasive and Mobile Computing, Elsevier, 2022. (DOI: 10.1016/j.pmcj.2022.101726).
- Riccardo Presotto, Gabriele Civitarese, Claudio Bettini, *"FedCLAR: Federated Clustering for Personalized Sensor-Based Human Activity Recognition"*. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications (PerCom), 2022.

My Contributions:

- Problem identification and formulation.
- Collaboration in methodology design.
- Design of the evaluation methods.
- Experiments execution.
- Results analysis and interpretation.

1.2.4 Investigating The Potential Privacy Issues In FL HAR

Even though FL avoids the release of labeled sensory row data, the parameters of deep learning models shared between the users and the cloud server may still reveal some sensitive information through specifically designed attacks [26]. This problem is particularly relevant considering the HAR domain where the involved information includes sensitive data regarding the users' health state and habits.

In Chapter 6 we propose the first contribution in this line of research by introducing a novel methodology to evaluate the effectiveness of the Membership Inference Attack (MIA) for FL-based HAR.

Our preliminary results on a public dataset suggest that the global activity model may reveal sensitive high-level information from participating users, hence providing hints for future works on countering such attacks.

Chapter 6 is based on the following publications:

- Riccardo Presotto, Gabriele Civitarese, Claudio Bettini. *"Preliminary Results on Sensitive Data Leakage in Federated Human Activity Recognition"*. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications Workshops, 2022.

My Contributions:

- Collaboration in methodology design.
- Collaboration in system implementation.
- Collaboration in the design of the evaluation methods.
- Collaboration in experiments execution.
- Collaboration in results analysis and interpretation.

1.3 Outline

The rest of the thesis is structured as follows. Chapter 2 provides a wide overview of the state-of-the-art for sensor-based activity recognition field, and introduces the specific challenges tackled by this thesis. Chapter 3 presents a novel hybrid activity recognition framework that relies on knowledge-based reasoning to refine the prediction of a machine learning classifier considering the context information. In Chapter 4, we introduce a novel semi-supervised and federated learning-based approach to reduce the data scarcity problem of HAR while preserving the privacy of the users. A valuable solution to tackle the non-IID problem typical of federated learning-based solutions for HAR is then illustrated in Chapter 5. In Chapter 6, we describe a novel framework to quantitatively measure the potential information leakage of the model parameters shared in federated HAR. Lastly, Chapter 7 summarises our contributions, outlines future works, and concludes the thesis.

Chapter 2

Related work

2.1 Human Activity Recognition

Human Activity Recognition is an enabling technology for the next generation of pervasive systems and context-aware services as it allows obtaining information about people, such as their health state, their habits, or monitor their dysfunctional behaviors [1]. HAR finds also useful applications in healthcare and medical systems, as it allows the development of advanced services such as remote patient monitoring and telemedicine. In recent years, the advent of ubiquitous computing and the widespread adoption of smart sensors and IoT devices in our everyday life enables collecting, storing, and processing of information related to human activities. Nowadays, the most common HAR systems rely on different typologies of data generated by a variety of devices and sensors [31, 32]. These include video-based HAR, sensor-based HAR, and wireless signals-based HAR. Video-based HAR analyses videos or images from cameras containing human motions [33, 34]. Wireless signal-based human activity recognition takes advantage of the signals propagated by wireless devices to classify human activities [35]. Differently, in sensor-based HAR the data are usually collected by inertial sensors embedded into wearable devices (e.g., smartphones and smartwatches) or ambient sensors dispatched in the living environment of the users (e.g., magnetic sensors, smart plugs). In the last few years, sensor-based HAR dominated the research landscape due to the ubiquity, unobtrusiveness, cheap installation pro-

cedure, and ease of usability of the devices involved in the data collection process [36]. Accordingly, in this thesis we will focus on sensor-based HAR.

2.1.1 Sensor-Based HAR

Overall, sensor-based HAR relies on the data collected from different sources: wearable and environmental sensors. The sensor equipment varies according to the typology of the considered activities. For example, physical activities like *walking*, *taking the stairs*, *or cycling*, are mostly characterized by the physical movement of the users. Therefore, there are usually considered only the data coming from the inertial sensors (e.g., accelerometer, gyroscope) embedded into wearable devices like smartphones or fitness bands. Differently, Activities of Daily Living (ADLs) like *cooking*, *taking medicine*, or *watching TV*, along with the body movement of the user, also implicate interactions with the living environment (e.g., open the medicines' door to take the medicine, and then close it). Therefore, in this case, it is more profitable to use both wearable sensors as well as environmental sensors (e.g., a magnetic sensor able to detect if the medicine door is opened or closed). Regardless of the adopted sensor equipment, sensor-based HAR solutions evolved by following a developmental pipeline with well-defined steps such as data collection, pre-processing, segmentation, features extraction, and finally, the training of a classification model through machine learning algorithms [37, 38]. Overall, the most common approaches for sensor-based HAR can be divided into two categories relying on how features are extracted and selected. On the one hand, we have standard machine learning techniques that assume that the features are handcrafted. On the other hand, deep learning-based approaches can automatically generate the features during the training of the classification model.

Standard Machine Learning Approaches For HAR

The general goal of HAR is to learn a machine learning model by minimizing the discrepancy between the model prediction and the ground truth activity. In order to achieve that goal, traditional machine learning approaches follow the four steps presented in Figure 2.1. First of all, the pre-processing step usually consists of

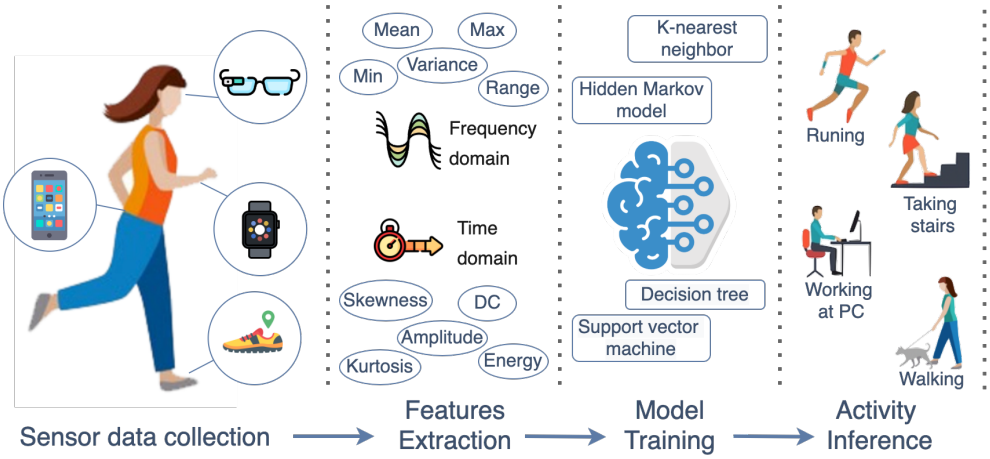


Figure 2.1: An illustration representing the pipeline adopted by standard sensor-based activity recognition approaches

filtering the raw sensor data in order to reduce noise. The pre-processed stream of sensor data is hence partitioned into segments of a specific length. Then, a set of features (e.g., *mean*, *variance*, *standard deviation*) are manually extracted from each of those segments based on human knowledge [9]. Lastly, the obtained feature vectors are used to train the machine learning model in order to classify activities.

Among the many different standard machine learning algorithms for HAR proposed in the literature, the most common are decision three [39, 40], Support vector machine (SVM) [41, 42, 41], K-nearest neighbors (KNN) [43, 44], and Hidden Markov Models (HMM) [45, 46]. Despite the promising results in terms of the recognition rate obtained from these approaches, the major disadvantages of standard machine learning bases solutions are related to the handcrafted feature extraction process. Indeed, in some cases, human experts are not able to select the best set of features [47]. Further, irrelevant features may be generated, making it necessary to apply methods that reduce the dimensionality of the data.

Deep Learning Approaches For HAR

In order to tackle the problems related to feature extraction, deep learning-based solutions for HAR have been proposed. Indeed, as Figure 2.2 illustrates, Deep

learning (DL) algorithms are capable of automatically generating features representing the raw sensory data while learning the recognition model. The most

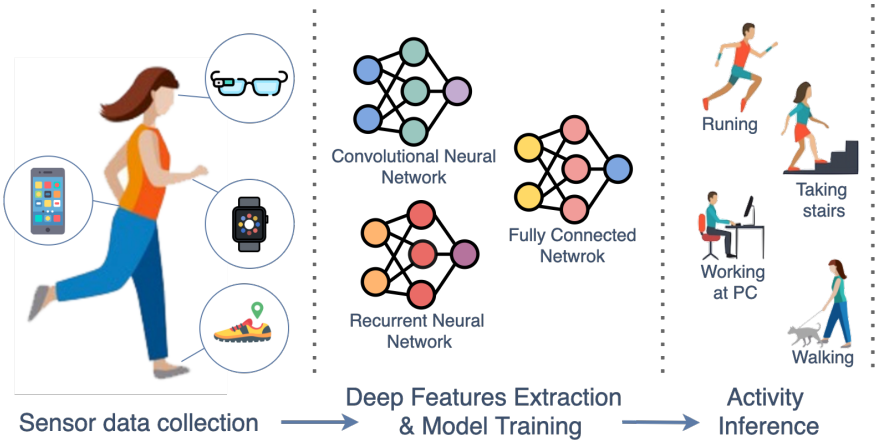


Figure 2.2: An illustration representing the pipeline adopted by deep learning based approaches for human activity recognition

common Deep learning-based approaches for sensor-based HAR are Convolutional Neural Networks (CNN), Recurrent Neural networks (RNN), and Fully Connected networks (FCN). In particular, CNN based-approaches showed to be more effective in terms of activity recognition rate with respect to both SVM [48] and Random Forest-based methods [8]. Considering RNN, the authors in [49] propose a HAR model based on Ensembles of deep Long Short Term Memory (i.e., LSTM is a specific type of RNN), which achieved a very good accuracy over the most common publicly available datasets for HAR [50, 51]. Moreover, it has been also proposed to use hybrid DNN models to identify human activities. For instance, the hybrid CNN and LSTM neural network presented in [32] overtakes the classification rate obtained by adopting CNN and LSTM networks separately. One of the major drawbacks of DL approaches is that they need a wide set of annotated samples to automatically compute the features while learning the recognition model. However, as we will detail in Section 2.1.1, the collection of such a large number of labeled data is a real challenge, especially considering the generation of a personalised recognition model.

Label Data Scarcity In sensor-based HAR

Considering sensor-based human activity recognition, the majority of the proposed machine learning based solutions rely on supervised learning approaches [51, 52, 53, 9, 54]. On the one hand, these approaches enable obtaining an excellent recognition rate but, on the other hand, they need a considerable amount of labeled data to train the classifier. Indeed, different users may perform the same activity in very different ways due to their physical characteristics, age, spot attitude, etc. Moreover, distinct activities may also be associated with similar motion patterns (e.g., sitting and standing). Thus, arise the need to collect a wide number of annotated samples from numerous users in order to train recognition models in a fully supervised way. However, in HAR scenarios the annotation task is very challenging as it is intrusive for the users, time-consuming, costly, and hence prohibitive on a large scale [12]. In the following, we summarize the most common methodologies that have been proposed in the literature in order to mitigate the annotated data scarcity problem.

Unsupervised approaches have been used to derive activity clusters from unlabeled sensor data [55]. Those approaches still need a few annotations to reliably associate an activity label to each cluster. Since distinct human activities often share similar sensor patterns, purely unsupervised data-driven approaches for activity recognition are still a challenge considering real-world scenarios. Data augmentation is one of the most popular solutions adopted in the literature to mitigate the data scarcity problem, especially given imbalanced datasets [56, 57]. In these approaches, the available labeled data are slightly perturbed to generate new labeled samples. Recently, data augmentation in HAR has also been tackled taking advantage of GAN models to generate synthetic data more realistic than the ones obtained by the above-mentioned approaches [58, 59]. However, GANs require to be trained with a significant amount of data. Then, many transfer learning approaches have been applied to HAR to fine-tune models learned from a source domain with available labeled data to a target domain with low-availability of labeled data [60, 61, 62, 63].
In the last few years, self-supervised learning (SSL) has drawn attention from

the research community. Indeed, SSL provides a general framework for learning from unlabeled data through solving a pretext task. In particular, a surrogate objective (i.e., the pretext task) is specified in such a way that optimizing it would force the network to learn meaningful and usable features for the downstream task (e.g., classification). Once the pretext task has been solved, a simple fully connected layer for classification can be added on the top of the learned network and then trained using a small batch of labeled examples. By following this approach, several SSL-based solutions have been proposed, especially in the computer vision and natural language processing domains [64, 65]. In view of the encouraging outcomes obtained in these domains, SSL has been very recently applied also for sensor-based HAR, leading to interesting results [66, 67]. However, given the novelty these SSL-based HAR solutions, we can consider them as the first steps in a promising research direction that is rapidly evolving. Among

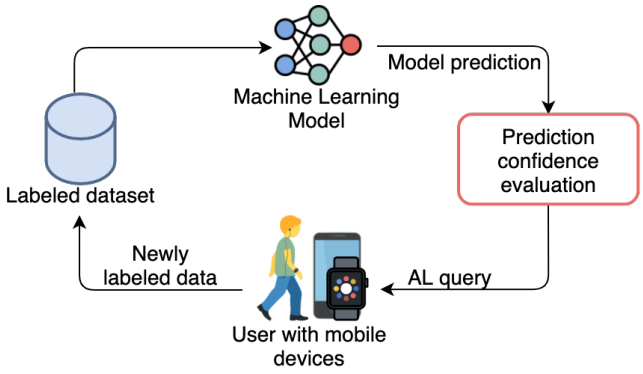


Figure 2.3: An example of the active learning cycle

the various methodologies proposed to address the annotated data scarcity problem in HAR, semi-supervised learning based approaches seem to be the most effective and mature [13, 68, 69, 70]. Indeed Semi-supervised methods only use a restricted labeled dataset to initialize the activity model. Then, a significant amount of unlabeled data is semi-automatically annotated. The most common semi-supervised approaches for HAR are self-learning [68], label propagation [71], co-learning [72], and active learning [14, 73, 16, 15]. Active learning has also been adopted in HAR to handle the class imbalance problem [74]. Figure 2.3 shows an example of the active learning (AL) cycle used to annotate unlabeled data. However, existing semi-supervised solutions do not usually consider the scalabil-

ity and privacy problems that may arise in generating a recognition model with a large number of users for a real-world deployment.

2.1.2 Knowledge-Based Methods

The information about the context that surrounds the user (e.g., semantic location, weather condition, time of the day, etc.) can be used to significantly extend the set of considered activities and to better discriminate the ones with similar motion patterns that are generally executed in different context conditions (e.g., sitting and sitting on a bus) [29]. However, the acquisition of a comprehensive training set where activities are performed in all the possible context conditions is prohibitive. The abstraction ability of common-sense knowledge can be used to generate formal models representing the relationships between context and activities [18]. Several approaches have been proposed in the literature to formally represent context data [75]. Ontologies have been preferred over other formalisms for activity recognition mainly for their expressive power and automatic reasoning capabilities [76, 77, 78]. There are several well-known ontologies that propose a formalism for context and activities, like *SOUPA* [79], *MetaQ* [80], and the so called *foundational ontology* [81].

2.1.3 Hybrid Machine Learning And Knowledge Based Methods

The combination of ontological context reasoning tools and machine learning algorithms on sensor data has been also explored. Banos et al. [82], proposed the integration of machine learning, used to derive low-level activities, with ontological reasoning, used to infer higher-level context based on the derived activities and other context sources (e.g., mood, semantic location). Ontological reasoning has also been used to integrate context data derived from machine learning processes in complex industrial IoT scenarios [83]. This is a typical application of ontologies, particularly useful when data is gathered by different sources and organizations. Moreover, some other hybrid methods propose to refine the prediction of a machine learning classifier by using knowledge-based reasoning over

context data [18]. Indeed, semantic reasoning can exclude from the prediction those activities that are highly unlikely according to the current context. The major issue of those hybrid solutions is the rigidity of the ontological formalism that is based on logical rules that despite offering the power of abstraction, struggle to capture the probabilistic nature of the relationship between context and activities. Overall, there is a vast literature on logic formalisms that support some form of uncertainty reasoning. Some efforts have also been made specifically for applications in the area of pervasive computing [84]. Considering description logics as the underlying logics of ontologies, an integration with fuzzy logic has been proposed to express confidence values for each axiom [85].

One of the well-known formalisms that combine logic with probability theory is *Markov Logic Network* (MLN) [86]. MLN can model both hard and soft constraints using weights associated with each rule. Generally, the weights associated with soft constraints are learned from labelled data. MLNs have been proposed for smart-home activity recognition [87]. However, they are less suitable than ontologies to model the complex hierarchical relationships between context data and activities. More recently, *probabilistic ontologies* have been proposed. Examples of such ontologies are *PR-OWL* [88, 89], *DISPONTE* [90, 91] and *Log-linear Description Logics* [92].

2.2 Collaborative Learning For HAR

One of the biggest challenges in the field of HAR is to train a global recognition model over the data coming from a large number of different users in a scalable and privacy-preserving way. However, as Figure 2.4 illustrates, the majority of existing approaches in the literature involve centralizing the users' data on a single server where the global recognition model is trained. This approach can create significant scalability challenges and privacy concerns. Firstly, centralizing data from a large number of users onto a single server can require a massive amount of computational resources and storage capacity. This can result in significant costs and resource requirements, which may make it difficult or even impossible for some organizations to implement. Secondly, centralizing the data on a single

server poses potential privacy risks. If the server were to be compromised, an attacker could gain access to sensitive user data, potentially resulting in negative consequences for the individuals involved. [93, 11]. A promising solution to

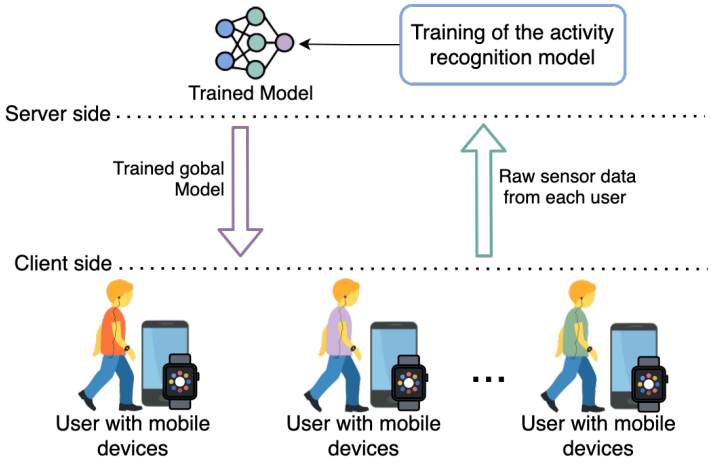


Figure 2.4: The architecture of a standard collaborative learning approach for HAR

reduce the privacy risks for the users in collaborative HAR, has been proposed in CollAR [94]. Here, the users cooperate to train a global model by sharing only a small portion of the parameters of their personal models. Moreover, the proposed framework enables the users to define which activities to keep private. However, its performance on large scale is still unclear in terms of communication latency and computational costs.

2.2.1 Federated Learning Approaches

To address both scalability and privacy concerns, the Federated Learning (FL) framework has been shown to be effective in various domains [95, 96, 97, 19, 98, 99]. FL offers a promising solution by enabling the collaborative training of a global recognition model without requiring data centralization on a single server. This approach enables distributed learning over multiple devices, preserving the privacy of users' data and reducing the computational and storage requirements for the central server.

In FL, the training of the global activity model is distributed among the

participating clients. Each client trains a local model with its available labeled data and only transmits local model parameters to the server, instead of sharing private activity data. The server aggregates the parameters received by the participating clients and creates a global model. Moreover, privacy-preserving mechanisms like Differential Privacy (DP) and Secure Multi-Party Computation (SMC) are usually adopted to further protect the shared model weights from attacks that can potentially reverse-engineer data or properties from the weights [95].

Federated Learning (FL) has recently been proposed as a solution for sensor-based HAR, allowing the collaborative training of a global recognition model among participating clients [100, 21, 101, 102, 23, 103, 20]. While FL solutions for HAR achieve slightly lower recognition accuracy than standard centralized models [21], personalization is essential for HAR [24]. To address this, existing works have applied transfer learning strategies to fine-tune the global model on each client, leading to significantly improved recognition rates [101, 23]. However, these solutions assume high availability of labeled data and a fully supervised setting.

Although federated and active learning have been combined for Intrusion Detection Systems [104], semi-supervised federated learning solutions for HAR are only partially explored. Existing works mainly focus on unsupervised methods to learn a robust feature representation from the unlabeled sensor data stream using FL, with the global feature representation used to build activity classifiers using a limited amount of labeled data. For example, [102] proposes an autoencoder-based approach, while [105] is based on self-supervised learning. However, these works do not consider model personalization or approaches for continuously obtaining new labeled data from users.

The Non-IID Problem

Another important aspect to consider in FL for HAR is that data coming from different users are likely non-independently and identically distributed (non-IID). Indeed, different subjects may have different physical characteristics and habits

and hence execute activities in very distinctive ways. Therefore, a single global model would lack in capturing those differences over a large number of users [106]. Several recent works proposed approaches to mitigate the non-IID problem in FL-based HAR. A common solution is to adopt transfer learning techniques on the client-side to improve personalization [101, 23, 103]. In particular, the global model is fine-tuned on each client by training the last layers of the personal deep learning model (i.e., the ones closest to the output) with personal data. The intuition behind this approach is that the last layers capture the personal patterns of the users, while the first layers encode cross-subject features [107]. However, those approaches are still based on a single global model, and balancing personalization and generalization is still a challenge. Multi-task federated learning is another approach proposed to mitigate the non-IID problem in FL-based HAR [108]. In particular, the clients contribute to collaboratively learning only the common features, while the diversity is handled on the client side. Nevertheless, these approaches are based on a convex objective function that is not suitable for complex HAR models based on deep learning.

A very recent work, that is called ClusterFL, proposes a multi-task federated clustering method for FL-based HAR [25]. This method is based on a distributed optimization approach: the clients and the cloud server collaborate in optimizing both the local models as well as the clustering structure. A limitation of ClusterFL is that the information about the association between users and clusters, as well as the parameters of each local model, are distributed to all the participating clients. Hence, ClusterFL does not adhere to the standard FL protocol and it reveals sensitive information to each client. Moreover, ClusterFL requires mobile devices to compute an optimization task that is more computationally expensive with respect to the usual local training required by FL approaches. The recent FL literature proposes Federated Clustering approaches that adhere to the standard FL protocol, without revealing clustering details to the participating clients [109, 110, 111].

Privacy Risks

The large amount of information that may be easily collected thanks to the potential of pervasive computing, may expose users to privacy risks, since most of this information may be private and sensitive. One of the challenges of Human Activity Recognition is to protect these sensitive data, while being still able to provide useful services and build effective activity classifiers. As we previously mentioned in Section 2.2.1, Federated Learning (FL) has been proposed as a privacy-preserving framework for distributed machine learning and recently, it has been also adopted in the HAR domain [21, 23, 25]. However, recent studies reveal that the model's parameters received and manipulated by the cloud server may still reveal information about the FL participating users. In the literature, the most investigated attack techniques are: a) the Membership Inference Attack (MIA) [27, 28, 112], b) the Property Inference Attack (PIA) [113], and c) the Reconstruction Attack (RA) [114, 115, 116]. MIA enables inferring if a given sample has been used or not to train a Deep Learning model (More details regarding the Membership inference attack are given in Section 6.2). Differently, the PIA technique aims at extracting properties of training data that may not be directly related to the task of the classifier (e.g., use the HAR model to infer if a subject suffers from the Parkinson disease). Finally, the RA technique has the objective of reconstructing prototypical examples of the samples used to train the machine learning model (e.g., reconstructing sensor data patterns that reveal sensitive physical characteristics of a subject). For instance, in a recent work [117] the authors show that given a Deep learning model trained over portraits of humans, it is possible to exploit RA to reconstruct pictures that look similar to the ones used to train the model. These attacks may be countered by adopting additional privacy-preserving mechanisms like Differential Privacy (DP)[118, 119] and Secure MultiParty Computation (SMC)[120]. However, those privacy-preserving mechanisms negatively impact the classification rate and system efficiency. Understanding and balancing those trade-offs is one of the major challenges in this area.

2.3 Datasets For Sensor-Based HAR

In the literature are available a large number of benchmark datasets for Sensor-based Human Activity Recognition [121, 122, 50, 51, 32, 123, 124, 125]. In the following of this section, we describe which of those datasets we consider in this thesis. In particular, we selected the ones that include data from a large number of heterogeneous subjects, as it is a fundamental aspect to evaluate the generalization and personalization capabilities of collaborative approaches for HAR. Furthermore, we introduce DOMINO, a self-acquired dataset that along with the inertial sensors data collected from wearable devices, includes a detailed description of the context in which activities have been performed.

2.3.1 MobiAct

MobiAct [121] includes labeled data from 60 different subjects with high variance in age and physical characteristics. The dataset contains data from a triaxial accelerometer, gyroscope, and magnetometer embedded into a smartphone carried by users while performing 9 physical activities in 16 trials. During the acquisition process, the users were left free to position the smartphone with a random orientation into one of their trousers' pockets. The physical activities included in this dataset are the following: *Standing*, *Walking*, *Jogging*, *Jumping*, *Upstairs*, *Downstairs*, *Sitting*, *Car step in*, *Car step out*. The data distribution of these activities is illustrated in Table 2.1. Regarding the participants' gender, 73% of them are male, while 27 are female. The subjects' age range between 20 and 47 (average: 26), the height ranged from 160 cm to 189 cm (average: 175), and the weight varied from 50 kg to 120 kg (average: 76). The adopted data acquisition frequency is the highest enabled by the sensors of the selected smartphone (i.e., at most $200Hz$).

2.3.2 WISDM

The well-known WISDM dataset [51] has been widely adopted as a benchmark for HAR. WISDM contains accelerometer data collected from a smartphone located in the front pants leg pocket of each subject during activity execution. The activ-

Activity	percentage of records
Standing	32.6%
Walking	37.8%
Jogging	7.8%
Jumping	7.4%
Sitting	5.5%
Upstairs	2.5%
Downstairs	2.3%
Car step in	1,7%
step out	1,9%
TOTAL	10,788,386 records

Table 2.1: MobiAct: distribution of the considered activities

Activity	percentage of records
Walking	38.6%
Jogging	31.2%
Sitting	5.5%
Standing	4.4%
Upstairs	11.2%
Downstairs	9.1%
TOTAL	1,098,207 records

Table 2.2: WISDM: distribution of the considered activities

ities included in this dataset are the following: *walking*, *jogging*, *sitting*, *standing*, *Upstairs*, and *Downstairs*. The distribution of activity classes in WISDM is illustrated in Table 2.2. While performing these activities, the sampling rate for the accelerometer sensor was kept at 20Hz. WISDM includes data from 36 subjects. The data collection was supervised by one of the WISDM team members to ensure the quality of the collected data. Further information about the participants like gender, age, and weight distribution is not publicly available.

2.3.3 PAMAP2

The benchmark PAMAP2 [50] was recorded in a scripted setting where 9 participants were instructed to carry out a total of 12 activities of daily living, covering domestic activities and various sportive exercises. In particular, the activity data were collected from three inertial sensors made up of triaxial accelerometers, gyroscopes, and magnetometers located on the subjects' ankle, chest, and wrist regions while they performed the following activities: *rope jumping*, *lying*, *sitting*, *standing*, *walking*, *running*, *cycling*, *nordic walking*, *ascending stairs*, *descending stairs*, *vacuum cleaning*, *ironing*. The sampling rate adopted for data acquisition was set to $100Hz$. The distribution of the activities considered in this dataset is described in Table 2.3.

Activity	percentage of records
rope jumping	2.5%
lying	9.9%
sitting	9.4%
standing	9.7%
walking	12.2%
running	5.1%
cycling	8.5%
nordic walking	9.7%
ascending stairs	6.0%
descending stairs	5,4%
vacuum cleaning	9,0%
ironing	12,2%
TOTAL	3,850,505 records

Table 2.3: PAMAP2: distribution of the considered activities

Activity	percentage of records
Standing	38.6%
Sitting	31.2%
Lying	5.5%
Walking	4.4%
Running	11.2%
Cycling	38.6%
Brushing teeth	31.2%
Stairs up	5.5%
Stairs down	4.4%
Elevator up	11.2%
Sitting on transport	11.2%
Standing on transport	11.2%
Moving by car	9.1%
TOTAL	3,924,200 records

Table 2.4: DOMINO: distribution of the considered activities

2.3.4 DOMINO

The DOMINO dataset contains context-aware HAR data collected from 25 different users wearing a smartphone and a smartwatch. The activities included in this dataset are the following: *walking*, *running*, *standing*, *lying*, *sitting*, *stairs up*, *stairs down*, *elevator up*, *elevator down*, *cycling*, *moving by car*, *sitting on transport*, *standing on transport* and *brushing teeth*. Those activities have been acquired in different contexts like *working at the office*, *going around in the city (Milan)*, *driving*, *using public transportation*, *cycling*, and *staying at home*. Table 2.4 summarises the distribution of the activity considered in DOMINO. Overall, we recorded almost 9 hours of labeled and context-rich sensor data ($\sim$ 350 activity instances). In particular, in our data collection setup, users carry a smartphone in their pants' front pocket and a smartwatch on the dominant hand's wrist. Dedicated Android applications run on those devices to enable the users to collect and self-annotate inertial sensor measurements and context data. The inertial sensor data is collected from the built-in accelerometer, magnetometer, and gyroscope of both the smartphone and the smartwatch. In particular, we considered the maximum sampling rate of such mobile devices (i.e., 200 Hz for the smartphone and 140 Hz for the smartwatch).

Context data is acquired by the smartphone considering embedded sensors as well as publicly available web services. Since context information does not change frequently, it was collected every 15 seconds. The considered embedded sensors are: the barometer measuring height variations, the luminosity sensor, the microphone to obtain the environment's noise level, and the GPS revealing the user's location and speed. Moreover, additional context information has been derived by combining the smartphone's built-in sensors with public web services. *Google's Places API*[1] provides the user's closest semantic places (e.g., university); *OpenWeatherMap*[2] supplies current local weather conditions (e.g., rainy), while *Transitland*[3] provides information about the public transportation routes and stops closest to the user. we also collect temporal context like the moment of the day (e.g., morning, afternoon, evening), the day of the week, the season, etc.

2.4 Research Problems Addressed By This Thesis

In this section, we outline the research questions tackled in this thesis. For each question, we introduce the research problem and indicate the specific chapter where the problem is addressed.

Q1) Can knowledge-based and data-driven approaches be combined to reduce the number of annotated samples required to collaboratively learn a HAR model?

The majority of the state-of-the-art solutions for collaborative HAR rely on supervised machine learning approaches. Anyway, the acquisition of annotated dataset of activities is costly and often unfeasible for the users. In order to overcome that issue, semi-supervised learning methods for HAR have been proposed. However, semi-supervised methods poorly perform in recognizing those activities

[1] https://developers.google.com/maps/documentation/places/web-service/overview

[2] https://openweathermap.org/

[3] https://transit.land/

characterized by similar movements (e.g., walking, taking stairs). A valuable solution to mitigate that issue may involve including the user context (e.g., semantic location, weather condition) as additional features in a machine learning process. Unfortunately, acquiring a comprehensive training dataset that includes annotated examples of activities executed in all the possible context conditions is extremely challenging, especially in semi-supervised settings.

In Chapter 3, we propose novel hybrid semi-supervised and knowledge-based frameworks for HAR that enable outperforming in terms of classification rate the state-of-the solutions for semi-supervised HAR, while dramatically reducing the interactions with the users required to collect annotated examples.

Q2) Can the Federated Learning Framework be a valuable solution to implement a scalable and privacy-aware HAR system in a data scarcity scenario?

Despite the novel approach proposed in Chapter 3 enabling the classification of a wide set of activities while reducing the data annotation efforts for users, there are still several challenges that limit the deployment of these solutions in realistic scenarios. Scalability and privacy issues arise when collaboratively training a recognition model with potentially sensitive data from a large number of different users. Additionally, as the volume of data increases, it becomes increasingly challenging to store and process it on a single machine.

To tackle these limitations, in Chapter 4, we propose an innovative semi-supervised and federated learning-based approach that enables privacy-aware and scalable activity recognition while also considering the labeled data scarcity problem. Federated learning is a machine learning paradigm that enables training models on decentralized data sources without the need to collect and centralize them. Instead, each user's device trains the model locally using their private data, and only the model updates are exchanged among the users. This approach increase the privacy level by ensuring data ownership while enabling the training to scale over a large number of users.

Furthermore, in contrast with most of the state-of-the-art FL systems that assume that labeled datasets are available for each client, our approach combines active learning and label propagation to semi-automatically provide pseudo-labels to the unlabeled data stream. This solution enables each client to contribute to the model's training process, even if the labeled data is scarce. Moreover, clients can learn from the model's global knowledge while leveraging their own private data to improve the model's accuracy.

Q3) How is it possible to mitigate the non-IID problem typical of FL-based approaches for HAR?

In the literature, most of the FL-based approaches assume that the data collected by the users are independently and identically distributed (IID data). Unfortunately, in the field of HAR, that assumption cannot be always satisfied as the participating users may have different habits and/or perform the same activity in different ways depending on their physical traits. For instance, consider the activity of walking, which is a common daily activity for people of all ages. However, the way that younger individuals walk is often different from that of elderly individuals. Younger individuals tend to have a faster and more energetic walking style, while elderly individuals often walk at a slower pace and with less stride length. As a result, the data collected from young individuals and elderly individuals while performing the walking activity may differ a lot. This highlights the need for personalized models that can adapt to the differences in activity patterns between different groups of users.

Therefore, in Chapter 5 we propose a novel Semi-Supervised Federated Clustering method for Personalized Sensor-Based Human Activity Recognition. Our approach mitigates the non-IID problem by assigning a specialized classifier for each group of users that exhibit similar ways of performing activities. By clustering users based on the similarity of their model updates, we can create groups of users who perform activities in a similar manner. Each group of users is then assigned a specialized classifier that is trained on their data using a semi-supervised learning approach, which enables us to handle the labeled data scarcity prob-

lem. This approach provides each user with a customized model that accurately reflects their unique activity patterns, while still retaining the scalability and privacy benefits of the federated learning framework

Q4) Which sensitive information of the users could be inferred by a potential attacker that accesses the model parameters shared in an FL-based system for HAR?

Concerning the privacy aspects, it is important to note that the data used in sensor-based HAR can contain sensitive and private information about the participating individuals. For example, the data may reveal information about their health conditions, habits, or dysfunctional behaviours. If this information were to be accessed by an unauthorized party, it could potentially result in a breach of privacy for the individuals involved. While FL is a promising approach for protecting user privacy in distributed machine learning, recent studies have shown that, through specifically designed attacks, parameters of deep learning models may still reveal information about the data used for training [27, 28]. This raises concerns regarding the actual privacy level provided by FL-based systems for HAR.

To explore this issue further, in Chapter 6 we propose a methodology for evaluating which sensitive information could be inferred by a potential attacker that accesses the model parameters shared in an FL-based system for HAR. By understanding the types of information that are at risk of being leaked, we believe that it is possible to develop more effective privacy-preserving mechanisms to protect user data in FL-based HAR.

Chapter 3

Using Context Data To Mitigate The Data Scarcity Problem In Collaborative HAR

3.1 Introduction

One of the major drawbacks of fully-supervised machine learning solutions for HAR is the cost of collecting the amount of labeled data required to build a reliable recognition model [51]. Collaborative learning is a promising research direction, but still requires high effort from users to collect annotated training samples. Semi-supervised and incremental approaches have been proposed to overcome this issue, as they only require a small amount of training data to initialize the recognition model [15, 16, 68, 69]. However, their effectiveness on complex and context-dependent activities is still unclear. Considering also the context surrounding the user (e.g., semantic location, weather, traffic condition, speed, etc.) could be valuable solution to tackle this problem. [17, 18]. Nonetheless, directly using context data as additional features in the machine learning process may not be as effective as expected, given the large and dynamic number of context variables involved.

In this Chapter, we consider these problems and we propose a novel collabora-

tive activity recognition framework that combines semi-supervised learning and context-aware reasoning. An incremental machine learning classifier is in charge of inferring from inertial sensor data a candidate probability distribution over the possible activities. A knowledge-based reasoning engine is then used to refine the probability distribution considering context data. The system provides as output the most likely activity from the resulting context-refined probability distribution. Then, by following the semi-supervised approach, when the system is not sufficiently confident about the current activity despite the context-based refinement, it starts an active learning process: asks the user about the activity being performed and uses the answer to provide a new labeled sample to the incremental classifier. In particular, we propose and evaluate two different typologies of knowledge-based reasoning engines. The first one relies on a *deterministic ontology* and enables excluding from the statistical prediction vector the activities which are highly unlikely considering context data. The latter uses a *probabilistic ontology* that allows refining the statistical prediction vector by taking into account the intrinsic uncertainty that characterizes the relationships between activities and context data.

Our experimental evaluation shows that both of the proposed knowledge-based reasoning engines are effective in a) improving the recognition rate, b) extending the set of recognizable activities, and c) triggering a significantly lower number of active-learning queries. Moreover, the obtained results highlighted that the use of a probabilistic ontology enables to more realistically capture the complex relationships between activities and the context, leading to better results with respect to the deterministic knowledge-based approach.

The rest of the chapter is structured as follows. In Section 3.2 a detailed system overview has been introduced. Section 3.3 describes how we implemented our incremental human activity recognition statistical classifier. Section 3.4 presents the probabilistic ontology, the deterministic ontology, and the related knowledge-based reasoning engines. Then, in Section 3.5 the prediction confidence evaluation and the active-learning strategies are described. The proposed experimental evaluation and the obtained results are presented in Section 3.6. Finally, in Section 3.7 we introduce a practical demonstration of the proposed solution.

3.2 System Overview

In this section, we describe the overall architecture of the proposed approach, as depicted in Figure 3.1.

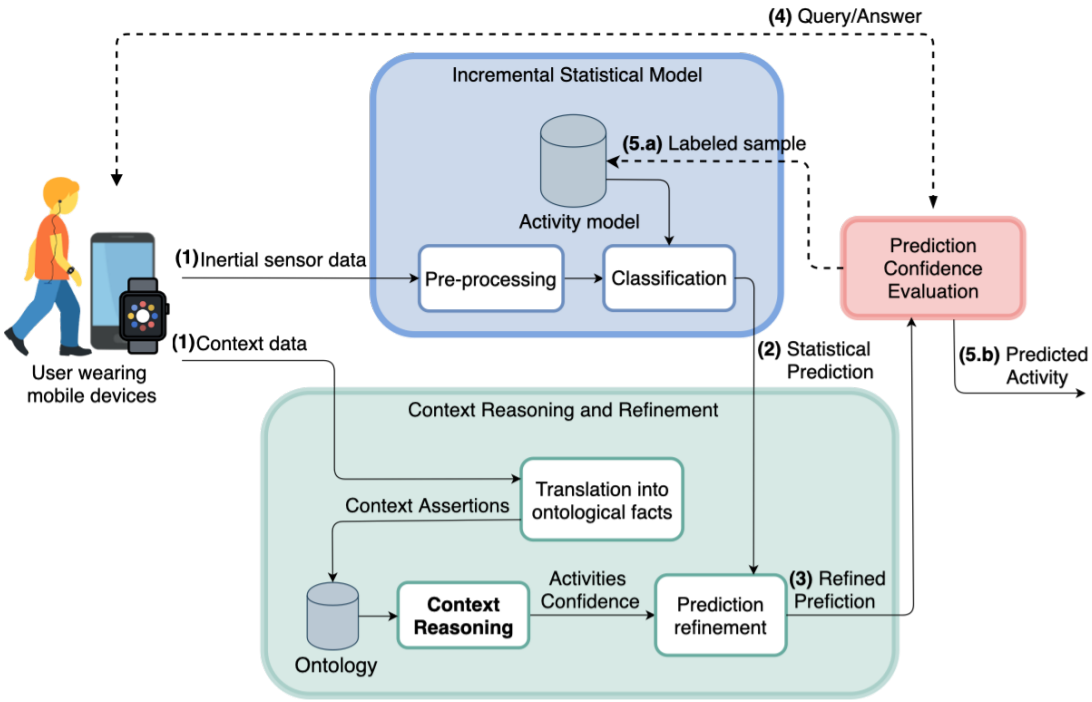


Figure 3.1: Overall architecture of our system

Inertial sensors data (e.g., accelerometer, gyroscope, and magnetometer) coming from multiple mobile devices are processed by the *Incremental Statistical Model* that relies on an incremental semi-supervised classifier to produce a candidate probability distribution over the possible activities. The mobile devices also dynamically acquire context data both by exploiting built-in sensors (e.g., GPS, luminosity sensor) and by querying publicly available Web services (e.g., Google APIs to obtain the user's semantic location). Note that, in the literature, "*context*" is a very broad term used to define a situation at different levels of abstractions [126]. For the sake of this thesis, with context data we denote the information about the environment that surrounds the user (e.g., user's semantic location, temperature, the time of the day) while the user is interacting with the system. Context data are analysed by the *Context Reasoning and Refine-*

ment module. This module applies knowledge-based reasoning on context data to refine the statistical prediction produced by the incremental statistical model. Finally, the *Prediction Confidence Evaluation* module adopts an active learning strategy based on a dynamic threshold. When the confidence over the refined prediction is lower than this threshold, a query is triggered to the user in order to obtain the ground truth. Upon receiving usable feedback, the module sends a new labeled sample to the incremental statistical model. In the following, we describe in more detail the components of our approach.

3.3 Incremental Human Activity Recognition

The *Incremental Activity Recognition* module relies on an online semi-supervised classifier to derive a candidate set of activities performed by the user. The stream of inertial sensor data is continuously pre-processed and segmented to extract feature vectors. These feature vectors are then provided to the classifier in order to derive the probability distribution over all the possible activities. Note that the activity recognition model is first initialized during an offline phase with a small amount of labeled data.

3.3.1 Segmentation, Feature Extraction, And Classification

In the following, we describe the pre-processing steps applied to the inertial sensors data stream. Since a user may carry multiple mobile devices (e.g., a smartphone and a smartwatch), it is first necessary to temporally align their raw sensor data streams. In our experimental setup, we considered for each device the data streams from the accelerometer, magnetometer, and gyroscope. A median filter is then applied to each stream in order to reduce the intrinsic noise of the signal. Then, the streams of aligned sensor data are segmented into segments defined as the set of inertial sensor data acquired during a specific time window of k seconds. Each segment starts the next second with respect to the end of the previous segment, hence segments are contiguous and non-overlapping. The length k is the same for all segments, and it should be chosen carefully according to the

complexity of the considered activities to balance the trade-off between accuracy and reaction time. In our experimental setup, we studied the existing literature to choose a reasonable fixed value for k [127]. Since our target activities are both simple (e.g., standing) and complex (e.g., driving) we could not choose a too-short window size. Hence, to guarantee a reasonable trade-off between accuracy and reaction time, we decided to use $k = 4$.

From each segment, a wide set of statistical features are extracted. These features have been selected from the ones that are well-known in the activity recognition literature [1]. In particular, for each axis of each inertial sensor, we extract: *average*, *variance*, *standard deviation*, *median*, *mean squared error*, *kurtosis*, *symmetry*, *zero-crossing rate*, *number of peaks*, *energy* and *difference between maximum and minimum*. Finally, for each inertial sensor, we compute the *Pearson correlation* for each combination of its axes and the *magnitude* on all of its axes. Hence, given q 3-axis inertial sensors equipped in the user's mobile devices, we compute $q \times 37$ features. We also apply standardization to each feature to further improve the recognition rate [128].

Example 3.3.1 *Consider a user which carries a smartphone and a smartwatch, both equipped with a three-axial accelerometer, gyroscope, and magnetometer. Since the overall number of inertial sensors is* 6*, for each segment,* $6 \times 37 = 222$ *features are computed.*

For each feature vector fv computed from a segment s, the incremental classifier h outputs a probability distribution over the set of considered activities $\mathbf{A} = \{A_1, A_2, \ldots, A_n\}$:

$$h(fv) = \langle p_1, p_2, \ldots, p_n \rangle$$

where $0 \leq p_i \leq 1$ is the probability $P(A_i|s)$ that the segment s was generated by the activity A_i, with $\sum_i^n p_i = 1$, and $n = |\mathbf{A}|$. The probability distribution $h(fv)$ is forwarded to our *Context Reasoning and Refinement* module which will refine it based on context data.

3.3.2 Activity Model Bootstrap

A crucial aspect of our semi-supervised framework is the activity model initialization. Indeed, without a proper bootstrap mechanism, the semi-supervised model would have to discover each activity "on-the-fly", with a negative impact on the recognition rate. Hence, we initialize the semi-supervised model by acquiring t seconds of labeled data for each activity to obtain a balanced labeled dataset. While the value of t should be as small as possible to minimize the effort in the labeled set collection, this value has a high impact both on the recognition accuracy and on the number of queries triggered to the users. In our experimental setup, we consider $t = 60$ (i.e., one minute) and hence, given the length of each segment, the activity model is initialized by using 15 labeled feature vectors for each activity. Based on our experiments and the range of considered activities, we believe that this small value should be sufficient to initialize an activity model that can rapidly evolve thanks to active learning.

3.4 Ontological Models

The *Context reasoning and Refinement* module is in charge of refining the prediction $h(fv)$ obtained by the *Incremental Activity Recognition* through the analysis of the context which surrounds the user. In order to achieve this task, this module relies on an ontology that models the relationships between context and activities. In particular, in this thesis, we implement and evaluate two different types of ontologies: a *deterministic ontology*, described in Section 3.4.2, and a *probabilistic ontology* illustrated in Section 3.4.3. The use of deterministic ontologies is the most common in the HAR literature because of their ease of implementation and usage. However, they involve defining rigid rules to model the relationship between the considered ontological concepts. Differently, probabilistic ontologies are more sophisticated and require a higher implementation effort, but they enable to finely model the relationships between activities and context.

3.4.1 Translating Context Data Into Ontological Facts

Context data collected by the mobile devices are automatically mapped to the respective ontological concepts by a specifically designed middleware. This middleware encodes the rules that are necessary to transform raw context data into high-level axioms. The majority of context data can be mapped one-to-one with ontological entities. For example, the user semantic location obtained from a dedicated service is directly mapped by the middleware to the respective ontological fact.

Considering scalar values, the middleware discretizes them taking into account the entities covered by the ontology. For instance, each user's speed value is mapped to one of the following ontological concepts: `NullSpeed`, `LowSpeed`, `MediumSpeed`, and `HighSpeed`. The specific rules that map a scalar value to an ontological concept are based on ranges of values designed by the knowledge engineer (e.g., speed values greater than 0 km/h and lower than 8 km/h are mapped to `LowSpeed`).

3.4.2 Deterministic Ontology

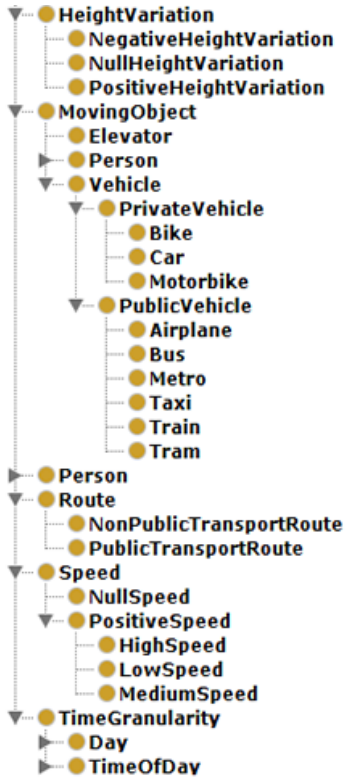
The proposed *deterministic ontology* is an extension of the *ActivO* ontology [18] that defines a wide set of activities, semantic locations, artifacts (e.g., used by the user or part of the semantic locations), user's postures, time granularities (e.g., day of the week, time of the day) and environmental information (e.g., temperature and light conditions). Details about*ActivO*'s implementation can be found in [18]. We took advantage of the Protégé tool [1] to extend *ActivO* with several new activities, contextual data, and their relationships. An example of those entities is shown in Figure 3.2.

Deterministic Ontology Modelling

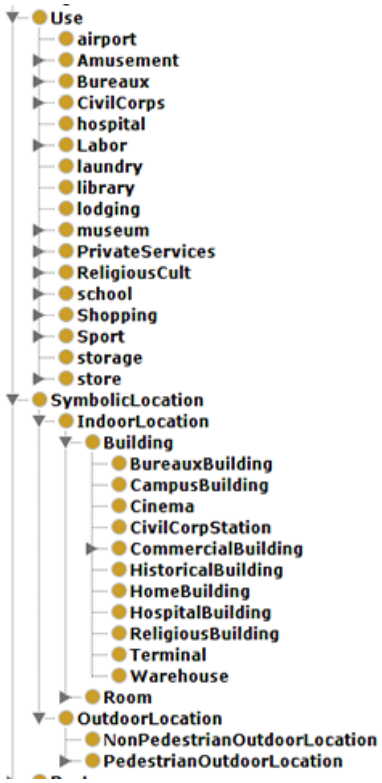
Our ontology considers several sources of context data: *user's semantic place*, *user's recent route*, *weather conditions*, *proximity to public transportation stops and routes*, *surrounding traffic condition*, *user's height variations*, *user's speed*,

[1] `https://protege.stanford.edu/` (Accessed on 2020-02-19)

surrounding light, *environment's noise level* and *temporal context (e.g., time of the day, day of the week, month, ...)*. Figure 3.2a shows a portion of those context data modeled in our *deterministic ontology*, while Figure 3.2b focuses on the set of considered semantic locations, including the ones classified by Google Places API [2]. It is important to note that we distinguish symbolic locations and their characteristics from their use. This allows us to better model activities related to symbolic locations.



(a) An excerpt of context hierarchy



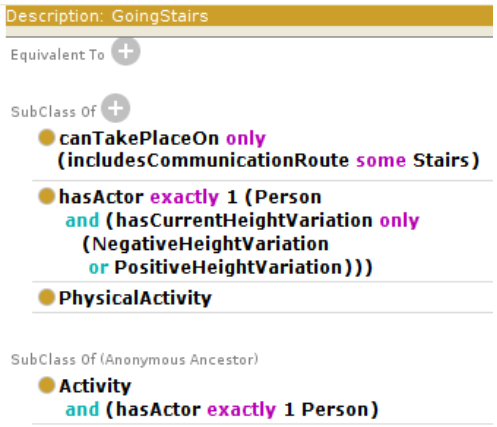
(b) An excerpt of symbolic locations hierarchy

Figure 3.2: Excerpts of our ontology

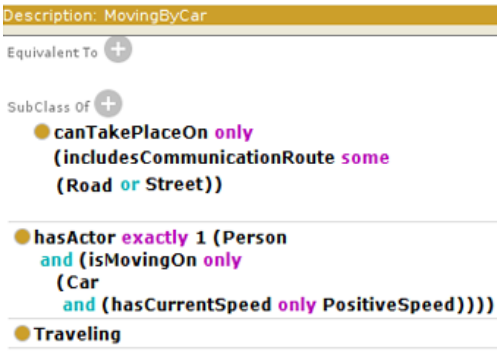
Due to the intrinsic open-world assumption of ontological reasoning, we explicitly state the necessary conditions which make activities possible or not possible in a given context. As we will explain later, such constraints are necessary to enable our context-aware refinement which is based on *consistency* reasoning. For

[2] `https://developers.google.com/places/supported_types` (Accessed on 2020-02-19)

instance, the activity `TakingStairs` (Figure 3.3a) should take place at a location that may have stairs and the person should have a non-negative height variation. Another example is the activity `MovingByCar` (Figure 3.3b): our *deterministic ontology* enforces that it should take place in an outdoor location which includes a road or a street and that the car's speed should be positive.



(a) Definition of the activity "taking stairs"



(b) Definition of the activity "moving by car"

Figure 3.3: Examples of activity definitions in our ontology

Deterministic Context Reasoning And Refinement

For each activity candidate $A_i \in \mathbf{A}$ in the statistical prediction $h(fv)$, our system uses *deterministic ontological reasoning* to determine whether A_i is consistent or not with the current context. As a first step, our system adds to the knowledge base an axiom to represent an instance of `Person` which identifies the subject wearing mobile devices. As a second step, context data collected by the mobile devices are automatically mapped to ontological concepts as described above and then added as axioms to the knowledge base. As a third and final step, since we want to test the consistency of activity A_i with respect to the current context, the system adds an axiom that states that the user is performing A_i.

Example 3.4.1 *Bob is using the system working with a deterministic context reasoning module. When the context reasoning task is triggered,* `Person(Bob)`

is added as a fact. Then, context data gathered by mobile devices is analyzed to expand the set of facts. Suppose that a Web service provides the information that Bob is in a park and that the speed value obtained by the GPS sensor is 10 *km/h. Those context information are automatically pre-processed to instantiate the following individuals:* `Park(place)`, `MediumSpeed(speed)`. *Then, the relationships between Bob and context data are added as facts:* `hasCurrentSymbolicLocation(Bob, place)`, `hasCurrentSpeed(Bob, speed)`. *Finally, in order to test whether the activity* `Running` *is context-consistent, CAVIAR adds the axioms* `Running (currentActivity)` *and* `isPerforming(Bob, currentActivity)`. *The consistency of the set of facts with respect to the domain knowledge will determine if the running activity is consistent according to the current Bob's context.*

We define an activity A_i *context-consistent* when the axioms created with the observed data as described above are consistent concerning the domain knowledge. Note that the consistency check involves reasoning that is automatically performed in the logic used to specify the *deterministic ontology*.

Given the current context C and the marginal probabilities obtained by the semi-supervised classifier $h(fv) = \langle p_1, p_2, \ldots, p_n \rangle$, the goal of the deterministic context refinement is to exclude those activities which are not *context-consistent* according to C. For each activity class A_i such that $p_i > 0$, we compute its consistency according to context C as explained above. Each activity that is not *context-consistent* is removed from the probability vector. The refined vector is finally normalized to preserve the properties of a probability distribution. The output is a new refined probability distribution over the possible activities:

$$predictions = \langle P_1, P_2, \ldots, P_n \rangle$$

such that each A_i is a *context-consistent* activity according to C, $\sum_{i=1}^{n} P_i = 1$, and $P_i \in [0, 1]$. Note that an activity is usually not *context-consistent* when the constraints of the deterministic ontology are violated.

Example 3.4.2 *Continuing Example 3.4.1, suppose that Bob is actually running. According to the Incremental Activity Recognition classifier, the current*

probability distribution is 45% *cycling,* 40% *running,* 10% *walking, and* 5% *standing. Thanks to a dedicated Web service, it is possible to know that Bob is currently in a pedestrian area of the park where bicycles are not allowed. According to the deterministic ontology, cycling is not context-consistent since it should not be performed in pedestrian areas. Hence, the resulting deterministic context-refined probability distribution is* 73% *running,* 18% *walking, and* 9% *standing.*

The resulting context-refined prediction vector is then processed by the *Prediction Confidence Evaluation* module that we will introduce in Section 3.5.

3.4.3 Probabilistic Ontology

In this section, we introduce the *probabilistic ontology* that, differently from a *deterministic ontology*, takes into account the intrinsic uncertainty that characterizes the relationships between context and activities. As previously described in Section 3.4.1 the context data obtained from the mobile devices are automatically translated into ontological facts, which are then added to the *probabilistic ontology* as a description of the *current* context condition. Then, probabilistic reasoning is in charge of inferring, given the current context situation C, a confidence value $conf(C, A_i)$ for each activity $A_i \in \mathbf{A}$. Intuitively, $conf(C, A_i)$ estimates the *"semantic compatibility"* of A_i being performed by the user whose current context is C. Finally, these confidence values are used to refine the probability distribution $h(fv)$ derived from inertial sensors data.

Probabilistic Ontology Modelling

The proposed approach combines *soft* and *hard* constraints to model the relationships between activities and context. *Hard* constraints capture context conditions that should always be satisfied to consider a given activity as possible. For instance, `Walking` is an activity that requires the user to have a positive speed. On the other hand, *soft* constraints are useful to capture context conditions that are likely to occur when an activity is performed, but not necessarily they have to be verified; this can be captured by associating a certain degree of confidence to the axiom. Intuitively, the highest the confidence and the more value will have the

presence of that context for the likelihood of the corresponding activity to occur. For instance, it is more likely that the activity `Running` is carried out on a sunny day rather than on a stormy day. Hence, the confidence value associated with the soft constraint "*running can be performed on a sunny day*" should be high, while the one associated with the soft constraint "*running can be performed on a stormy day*" should be lower.

In order to implement the *deterministic ontology*, we modified the publicly available OWL2 *ActivO* ontology [18] into a probabilistic ontology based on log-linear description logic [129]. A log-linear description logic is characterized by a *CBox* (i.e., Constraint Box) defined as $C = (C^D, C^U)$, where C^D is a set of *hard* axioms and $C^U = \{(c_1, w_{c_1}), (c_2, w_{c_2}), ..., (c_n, w_{c_n})\}$ is a set of *soft* axioms. Each soft axiom c_i is associated with a real-valued weight w_{c_i}.

The inclusion of an axiom in C^D and C^U is mutually exclusive. C^D is also assumed to be coherent and consistent (i.e., it is not possible to derive inconsistencies). A log-linear description logic relies on a log-linear probability distribution over the *coherent* and *consistent* subsets of the CBox. Each subset of the CBox represents a world that, if coherent and consistent, is associated with a probability computed using the weights of its soft axioms. Incoherent and/or inconsistent subsets of the CBox are considered as impossible. More details about log-linear description logics can be found in [129].

In our probabilistic ontology, activities are explicitly grouped according to context conditions. Examples of these groups could be "activities that can be performed indoor" or "activities that can be performed at a positive speed". We refer to these groups as *activity characterizations*. Clearly, an activity may belong to more than one characterization. Characterizations provide an abstraction layer that improves the ontology readability. Moreover, characterizations can be used to define mutually exclusive sets of activities. This approach also makes it possible to easily add new context conditions by creating a new characterization and binding it with the desired activities. Figure 3.4 and Figure 3.5 show how characterizations are represented in our ontology.

Each characterization is modeled as an equivalence axiom which describes a specific context condition that an activity should satisfy. Therefore, each activity can be modeled in terms of *set membership* to specific context conditions by using

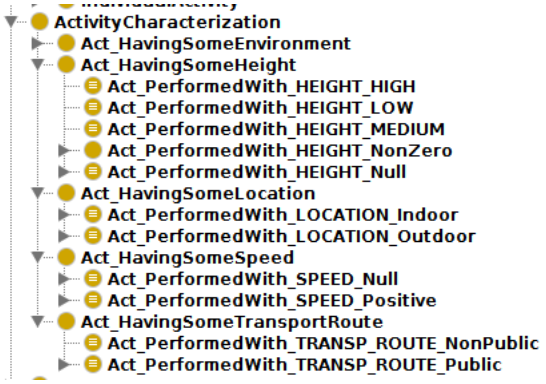


Figure 3.4: A subset of the characterizations in our ontology

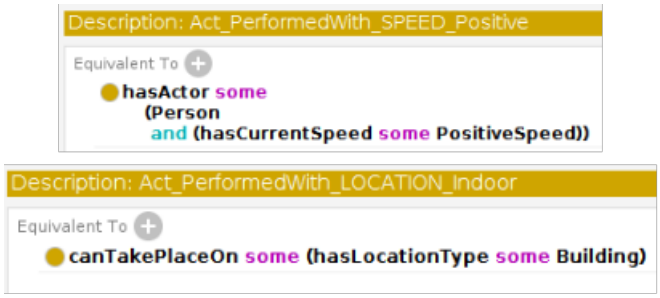


Figure 3.5: Examples of descriptions of characterizations in our ontology

subsumption or disjunction axioms with the characterization classes.

In order to understand how activities are modeled in our ontology, consider `Running`. It is clear that a person has to move with a positive speed in order to perform this activity. However, other context conditions related to *Running* should be modeled considering soft constraints:

- outdoor/indoor: even if it is more likely that a person is running outside, it is also possible to run inside a building;
- speed: a person may run with different speed rates and each rate has its own probability. Intuitively, a normal running speed rate is the most likely one for this activity, slow running (e.g., jogging) is slightly less probable, while running fast is the least likely one;
- height variation rate: a person may run on a flat or inclined road. Hence,

users may run at varying height variations. The most likely scenario is probably running on flat roads.

Due to these considerations, a possible probabilistic modeling of `Running` is depicted in Figure 3.6, using subsumption and disjunction relationships with the corresponding characterizations. The hard rules are recognizable by the absence of the yellow `OWLAnnotation` marker, which is enabled on the soft rules. Indeed, the specific log-linear logic that we adopted in our system associates a weight to each soft axiom by using an OWL2 annotation called *confidence*.

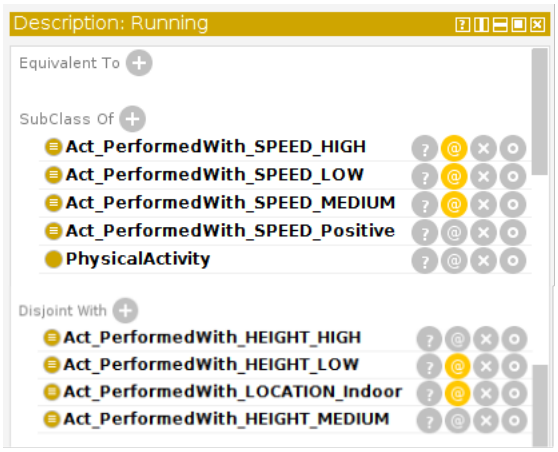


Figure 3.6: Description of *Running* using hard and soft axioms. The soft axioms are the ones associated with the yellow `OWLAnnotation` marker. Clicking on that marker it is possible to obtain the weight value.

Weighted subsumption axioms are used to describe uncertainty about the different values that a context condition can have. As we show in this example, our ontology includes a weighted subsumption for each possible speed rate related to `Running`.

Note that, for instance, the soft constraint of Figure 3.6 related to low speed can be formalized as follows:

$$Running \sqsubseteq Act\_Performed\_With\_SPEED\_LOW : w_1$$

where $w_1 \in \mathbb{R}$ is the weight associated with this axiom. Later in this section we will discuss how these weights are actually computed. The weight influences the veracity of other axioms related to the same context property (e.g. `Running` can be performed at medium/high speed rates). Therefore, when modelling weighted

axioms, we need to pay attention to how a specific weighted axiom influences the others in the reasoning process.

In our model, weighted disjunctions are used to represent uncertainty about context conditions considered both in hard rules and soft rules. For instance, if we model using a weighted subsumption axiom that `Running` may be performed indoor, then `Running` and other outdoor activities would be associated with different output probability values given the same context conditions. This would happen because of the semantics of log-linear DL, which would take into account also the indoor subsumption axioms during the reasoning process of the current user's context, which may specify that the user is outdoor. On the other hand, weighted disjunctions express a degree of incompatibility between activity and specific context information. In this case, the axiom would be taken into account during the reasoning process only if the current context contains that information. In this example, the weighted disjunction can be formalized as follows:

$$Running \sqcap Act\_Performed\_With\_LOCATION\_Indoor \sqsubseteq \bot : w_2$$

where $w_2 \in \mathbb{R}$ is the weight associated with the disjunction.

Axioms' Weights

In log-linear description logics, the weight associated with a soft axiom takes values in $\mathbb{R}$. In the literature, those weights are generally learned from labeled data. In our domain, the acquisition of a comprehensive annotated dataset that includes activities performed in a wide variety of context conditions is prohibitive. In this work, we associate with each axiom a probability value $p \in [0, 1]$ based on common-sense knowledge on context and activities. This knowledge should not necessarily come from the knowledge engineer and domain experts but it may be extracted semi-automatically in several ways, including:

- Proposing a survey to a large number of users;
- Scraping information about context and activities from the Web.

For example, suppose that, according to common knowledge, the activity *Running* is not very likely when performed in indoor environments. Hence, according to common-sense knowledge, our system associates the probability value 0.3 to

the soft axiom "running can be performed indoor", while 0.7 to the soft axiom "running can be performed outdoor".

Note that directly using probability values as weights associated with soft axioms is not a good choice given the underlying log-linear probability distribution. Hence, as proposed in other works [130, 131], we use the *logit* function to map each probability value p to a real number as follows:

$$\text{logit}\,(p) = \log\,(p) - \log\,(1-p) = \log\left(\frac{p}{1-p}\right)$$

The advantage of using *logit* is that it can approximate probability values for a log-linear model. Note that *logit* is not defined at 0 and at 1. When $p = 1$ or $p = 0$ we consider the axiom as a *hard* constraint. In the former case, it is a context condition that is always required for the corresponding activity; in the latter case, it is a context condition that should never occur.

Probabilistic Context Reasoning And Refinement

The proposed *Probabilistic Context Reasoning* engine uses the previously described probabilistic ontology to compute, given the current context data, a confidence value for each activity. First, context data is translated into ontological facts: class instances and relationships that populate the assertional part of the ontology. Once the ontology has been extended with facts about the current context conditions, it is processed by the probabilistic reasoner ELOG [129]. ELOG is in charge of computing marginal inference to obtain, for each activity A_i, a confidence value. Each confidence value $conf(A_i, C)$ estimates the compatibility of A_i with the current context condition C.

The marginal inference algorithm implemented in ELOG, called *MisSampler (Minimal Inconsistent Subset Sampler)*, analyzes the entire ontology to generate a posterior probability value for each soft axiom according to the log-linear description logic semantics. In order to compute a posterior probability value for each activity possibly performed by the user, our ontology includes dedicated soft axioms. In particular, there is an additional soft axiom for each activity. Each one of these axioms is declared as a subclass of the corresponding activity entity

with 0 as the default confidence value. According to the log-linear DLs semantics, the posterior probability of these axioms is 0.5 if they do not conflict with other axioms. Indeed, without conflicts, the posterior probability of an axiom c with weight w_c is defined by alogit(w_c) where alogit($\mathbb{R}$) $\rightarrow (0, 1)$ is the logit inverse function. Figures 3.7 and 3.8 show those additional soft axioms and their relationships with the rest of the ontology.

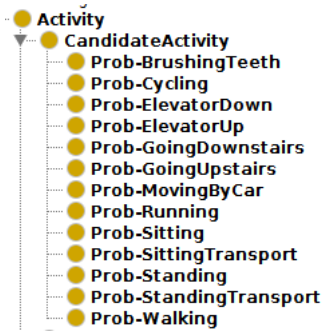


Figure 3.7: Probabilistic terminological overlay

Figure 3.8: CandidateActivity and Prob-Running classes

The output of the marginal inference algorithm is a vector of confidence values:

$$confidences(C) = \langle c_1, c_2, \ldots, c_n \rangle$$

where C is the input context data and $c_i \in \mathbb{R}+$ is the confidence value $conf(A_i, C)$ associated to the activity $A_i \in A$. Note that $conf(C)$ is not a probability distribution over the activities. Each c_i is a posterior probability value computed by the underlying log-linear probability distribution over coherent and consistent ontologies. Hence, these values should be considered as confidence values associated with the activities given the current context condition. Since we use the

default value 0 for the probabilistic axioms in the ontology, the value of each c_i is in the range $[0, 0.5]$ (due to *alogit* as we previously discussed).

Given a confidence value c_i:

- $c_i = 0.5$ reveals that the context satisfies the hard rules related to A_i without the involvement of probabilistic axioms;
- $0 < c_i < 0.5$ reveals that the context satisfies the hard rules of A_i but some soft axioms were used in the inference process, thus decreasing the output confidence. The more soft axioms are involved, the lower the output confidence;
- $c_i = 0$ reveals that the context does not satisfy at least one hard constraint for A_i. Therefore, according to our ontology, the activity is impossible in that specific scenario.

The confidence values inferred by ELOG are used to refine the probability distribution obtained from the *Incremental Statistical Model* on inertial sensor data. In particular, given the probability distribution $\langle p_1, p_2, \ldots, p_n \rangle$ and $\mathit{confidences}(C) = \langle c_1, c_2, \ldots, c_n \rangle$ such that A_i is an activity label, p_i is the probability associated to A_i by the statistical model and c_i is the ontological confidence value of A_i given the current context condition, we compute the following vector:

$$v = \langle p_1 * c_1, p_2 * c_2, \ldots, p_n * c_n \rangle$$

Hence, confidence values are used as weights associated with the probability values. Finally, the vector v is normalized in order to obtain a probability distribution over the possible activities:

$$\mathit{predictions} = \langle P_1, P_2, \ldots, P_n \rangle$$

such that $\sum_{i=1}^{n} P_i = 1$ and $P_i \in [0, 1]$. This probability distribution is the output of the Context reasoning module and is forwarded to the *Prediction Confidence Evaluation* module.

3.5 Prediction Confidence Evaluation and Active Learning

The *Prediction Confidence Evaluation* module is in charge of using context-refined predictions to update the activity model with newly labeled samples obtained through a custom active learning-based strategy.

3.5.1 Active Learning

In order to update and improve the activity model, we apply an active learning strategy asking a feedback from the user about her current activity when there is uncertainty in the context-refined prediction. In particular, we adopt a state-of-the-art non-parametric method called *VAR-UNCERTAINTY* [132]. This method is based on a threshold θ which is dynamically adjusted over time. Initially, this threshold is initialized to $\theta = 1$. Given a context-refined prediction $\langle P_1, P_2, \ldots, P_n \rangle$, we denote with $P^\star = \max_i P_i$ the probability value of the most likely activity $A^\star$. If $P^\star$ is below θ, we consider the system uncertain about the current activity performed by the user. In this case, an active learning process is started by asking the user to provide the ground truth A^f about the current activity. The feedback A^f is used to update the activity model with a newly labeled data sample. When $A^f = A^\star$, it means that the most likely activity was actually the one performed by the user, and hence the threshold θ is decreased to reduce the number of questions. On the other hand, when $A^f \neq A^\star$, θ is increased. More details about the VAR-UNCERTAINTY algorithm can be found in [132].

We assume that active learning queries are prompted to the user in real-time through a dedicated application, thanks to a user-friendly interface. Each query asks the user to choose the activity that she is currently performing among the possible ones. For the sake of usability, our system only presents a couple of alternatives taken from the most probable activities. Figure 3.9 shows a screenshot of the active learning application that we implemented for the smartwatch.

Figure 3.9: Illustration of our active learning interface for smartwatch.

3.6 Experimental evaluation

In this section, we describe how we evaluate the effectiveness of both the *deterministic* and the *probabilistic* context-based refinement approaches. Finally, we compare ad discuss the obtained results.

3.6.1 Deterministic Reasoning Based Results

To evaluate the effectiveness of the deterministic reasoning approach, we used the DOMINO dataset introduced in Section 2.3.4. This dataset contains inertial sensor data and context information regarding the execution of 14 different activities executed by 25 users wearing a smartphone and a smartwatch. The considered activities are the following: *walking, running, standing, lying, sitting, stairs up, stairs down, elevator up, elevator down, cycling, moving by car, sitting on transport, standing on transport, and brushing teeth.* These activities were recorded in various contexts, including working, going around in the city, and using public transportation. Then, we adopted Online Random Forest [133] as classifier, since it is the incremental version of the well-known Random Forest machine learning algorithm, which proved to be one of the most effective classifiers for activity recognition [122]. We take advantage of the Java implementation proposed in [134]. HermiT [135] in combination with the Java OWL API [136] is our OWL2 ontological reasoner. Since there is no system in the literature to

directly compare with, we implemented two variants of our approach. The former is called *Data-Driven approach*, since it only considers inertial sensor data to recognize activities. In particular, it combines the *Incremental Activity Recognition* module and the *Prediction Confidence Evaluation* module without applying our deterministic context-refinement approach. Note that *Data-Driven approach* can be considered as a baseline since it is a standard approach for activity recognition [1] The latter variant is called *Context as features*. This method, instead of using semantic refinement, incorporates context data directly in the feature vectors generated by the feature extraction mechanism presented in Section 3.3.1. In particular, this method extracts a) statistical features (average, variance, difference between max and min) from *numeric* context data like speed or height variations, and b) binary features for *symbolic* context data (i.e., semantic place, weather condition, proximity to transportation routes, etc.). We used a *leave-one-subject-out* cross-validation approach to evaluate and compare our approach with these two variants in terms of recognition rate and the number of subject questions. At each fold, we use 25 subjects to collaboratively update the activity model, which is initialized considering only 1 minute of labeled data samples for each activity. The data of the remaining subject is used to compute the recognition rate, and the number of questions asked to the subject.

Table 3.1 shows the results (in terms of overall F1 score). The results clearly show that context data has a significant impact on the overall recognition rate. Moreover, context data also allows our method to consider a wider set of activities compared to standard methods which only consider inertial sensors (e.g., *Without Context*). Indeed, activities that are characterized by similar inertial patterns but that are typically executed in very different context conditions can be easily discriminated by our deterministic context-based approach. For example, it is evident that activities like going upstairs/downstairs and sitting/standing on transport (which are more difficult to recognize only considering motion patterns) highly benefit from context data. On the negative side, we observe that the recognition rate of the deterministic context-based approach on the *cycling* activity is lower than the ones obtained by the other approaches. Indeed, this activity is often confused with *moving by car*. This is due to the fact that the available context data that characterizes those activities are similar (e.g., they

Activity	Data-Driven approach	Context as features	Deterministic Context Reasoning
Elevator up	0.0	0.09	**0.70**
Elevator down	0.31	0.65	**0.83**
Moving by car	0.76	0.80	**0.87**
Brushing teeth	0.77	**0.83**	**0.83**
Running	0.98	0.97	**0.98**
Sitting	0.94	0.96	**0.97**
Going upstairs	0.38	0.45	**0.77**
Going downstairs	0.58	0.81	**0.90**
Cycling	**0.96**	**0.96**	0.93
Standing	0.85	0.95	**0.96**
Walking	0.84	0.89	**0.95**
Sitting transport	0.35	0.62	**0.78**
Standing transport	0.41	**0.97**	0.90
Avg F1	0.62	0.77	**0.88**

Table 3.1: Recognition rate (F1-score) of the proposed deterministic context-based solution compared with alternative approaches

are both performed outdoor in the city traffic, with variable speed, etc.).

Besides the recognition rate, a crucial evaluation parameter is the number of questions triggered by the system, since it has a significant impact on usability. As Figure 3.10 shows, the proposed approach generates a significantly lower number of questions (6%) compared to *Data-Driven approach* (22%) and *Context as features* (16%).

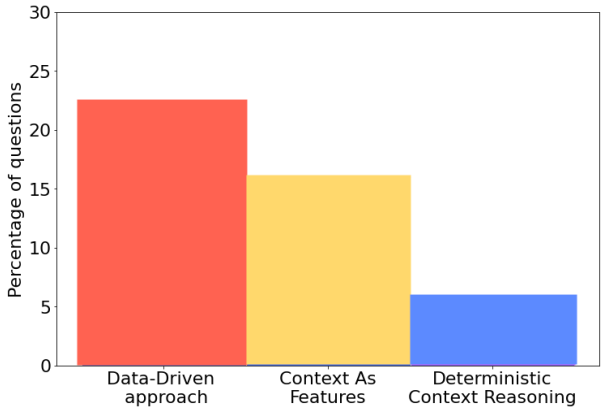


Figure 3.10: Percentage question triggered by the deterministic reasoning approach compared with alternative solutions

Indeed, our semantic deterministic refinement technique exploits the ontology to remove unlikely activities from the prediction, thus significantly increasing the confidence in the remaining activities. Accordingly, the experimental evaluation indicates that our approach reaches satisfying results in terms of classification rate using a very limited number of labeled samples, proving to be particularly appropriate in data scarcity scenarios like sensor-based HAR.

3.6.2 Probabilistic Reasoning Based Results

A possible limitation of the deterministic reasoning approach is the rigid formalism for knowledge representation and reasoning that cannot take into account the intrinsic uncertainty and incompleteness of common knowledge and sensor technology. However, the DOMINO dataset that we use to evaluate it does not include activities executed in context conditions that are unlikely but not impossible in realistic scenarios. For instance, the `Running` activity was never executed in indoor environments and/or with lower speed rates (e.g., jogging). Another example is the `Stairs up` and `Stairs Down` activities, which were never executed outdoor despite it is possible to find stairs outside. Since we want to quantitatively show if the probabilistic reasoning framework overcomes these drawbacks, we slightly modified the DOMINO dataset in order to incorporate unusual context scenarios.

Simulating Unusual Scenarios

We implemented a probabilistic simulator for context data which is based on the considered dataset. Hence, we replaced the original context data with simulated context data. For each activity class in the dataset, our simulator considers:

- context information which characterizes the activity regardless of the scenario (i.e., context data needed to satisfy the hard constraints of the ontology);
- a probability distribution over the context data that may be relevant for estimating the probability of the activity (i.e., context data captured by soft constraints in the ontology).

Our simulator relies on a probabilistic representation of the common knowledge of the activity domain to generate possible scenarios for each activity class. For each activity instance in the dataset, the simulator generates, based on the label, a scenario that includes context data related to hard constraints and some of the context data related to soft constraints. The latter are sampled from a probability distribution.

For instance, consider the activity `Walking`. Based on common-sense, our simulator incorporates the following probabilistic knowledge:

1. it is very common that users walk slowly (80% of probability), while they sometimes walk faster (20% of probability);

2. in the majority of the cases, users walk on flat surfaces, hence with no height variation (70% of probability), while they can walk ascending/descending paths with a lower probability (30% of probability);

3. *Walking* can be performed indoor or outdoor with equal probability.

For each activity instance, our simulator generates context data by sampling from these probability distributions. Continuing the example of `Walking`, a wide variety of context scenarios can be generated, like the following ones:

- **Scenario A:** {low speed, no height variation, indoor location}
- **Scenario B:** {low speed, positive small height variation, outdoor location}
- **Scenario C:** {medium speed, no height variation, indoor location}
- **Scenario D:** {medium speed, negative small height variation, outdoor location }

Intuitively, scenario A is the most common one for *Walking* and it would be frequently generated by our simulator. The other examples of scenarios are the least common, so they would be rarely generated by the simulator.

Probabilistic vs Deterministic Reasoning Results

In the following, we present the results obtained thanks to the proposed *probabilistic reasoning approach*. Here, we used the dataset presented in Section 2.3.4, enhanced with our probabilistic context data simulator. In order to evaluate the effectiveness of this technique we compare it with the *Data-driven approach* and the *deterministic context reasoning approach*.

We performed leave-one-subject-out cross-validation to assess the recognition rate of our system and the ones of the other approaches. Table 3.2 shows the results in terms of the overall F1 score[3].

Activity	Data-Driven approach	Deterministic Context Reasoning	Probabilistic Context Reasoning
Elevator up	0.0	**0.95**	**0.95**
Elevator down	0.27	**0.94**	**0.94**
Moving by car	0.78	**0.81**	**0.81**
Brushing teeth	0.77	0.77	**0.82**
Running	**0.98**	0.80	**0.98**
Sitting still	0.94	0.98	**0.99**
Going upstairs	0.50	0.63	**0.86**
Going downstairs	0.51	0.67	**0.83**
Cycling	0.95	0.95	**0.97**
Standing still	0.84	0.95	**0.97**
Walking	0.76	0.84	**0.94**
Sitting transport	0.31	0.86	**0.90**
Standing transport	0.48	0.94	**0.97**
Avg F1	0.63	0.86	**0.92**

Table 3.2: Recognition rate (F1 score) of probabilistic context reasoning compared with alternative approaches

The obtained results confirm that context data has a significant impact on the overall recognition rate. Most importantly, the *probabilistic context reasoning* approach significantly outperforms the *deterministic context reasoning* approach reaching an overall F1 score of 0.92. Indeed, thanks to its probabilistic perspective, it can recognize activities performed in unusual scenarios, considered

[3]Note that the results regarding *Data-driven approach* and *Deterministic Context Reasoning* exhibited in Table 3.2, differ from the ones presented in Table 3.1 as they have been obtained with the modified version of DOMINO which includes simulated unusual context scenarios. The same consideration is valid also for the other comparisons that we present in the following of this Section.

as impossible by the deterministic context reasoning approach. Looking closely at the results, some of the activities related to the highest improvements are *Going downstairs*, *Going upstairs*, *Walking*, *SittingTransport*, *StandingTransport* and *BrushingTeeth*. For these activities, our simulator generated a wide range of unusual scenarios, thanks to a higher number of combinations of context data with respect to other activities. Thus, the dataset contains more "unusual samples" for those activities with respect to the others, which are characterized by a smaller range of possible scenarios.

Considering the *deterministic context reasoning*, it is possible to observe a significant decrease in the recognition rate of `Running`. Indeed, while this activity can be reliably recognized only by analyzing inertial sensor data, the deterministic semantic refinement often considers it inconsistent considering unusual scenarios. For instance, the *deterministic ontology* described in Section 3.4.2 considers as impossible the fact that *Running* can be carried out indoor, since it is unlikely. Our method can overcome these problems thanks to soft axioms.

Besides the recognition rate, a crucial evaluation parameter is the number of questions triggered by the system, since it has a significant impact on usability. Figure 3.11 shows how both *Probabilistic* and *Deterministic reasoning* generates a significantly lower number of questions (respectively 6% and 8%) compared to *Data-Driven approach* (22%).

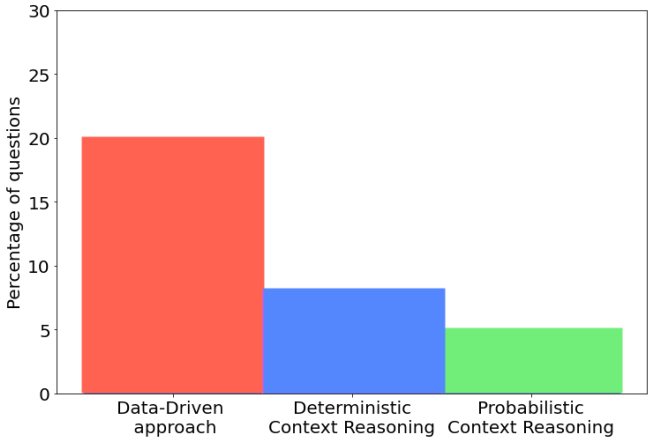


Figure 3.11: Percentage questions triggered by probabilistic context reasoning compared with alternative approaches

The *Probabilistic context reasoning* slightly decreases the number of user questions with respect to *Deterministic context reasoning*. Hence, on average, our probabilistic context refinement method further reduces the uncertainty on the output probability distributions compared to the deterministic solution.

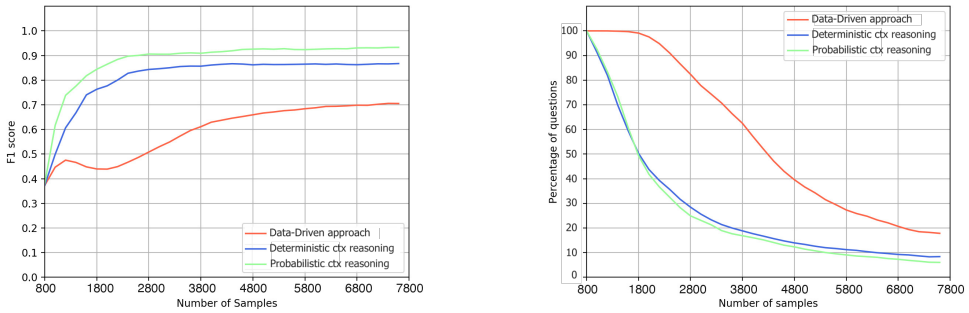
In order to evaluate how the recognition rate and the number of triggered questions evolve over time, we use the evaluation method proposed in [137]. We classify each data sample of the dataset (considering all 26 subjects) with the current model and, depending on the prediction's confidence, we update the recognition model. The classification's output (i.e., the most likely activity), and the corresponding ground truth are collected in sliding windows of 800 samples with an overlap of 75% to periodically compute the overall F1 score and the percentage of triggered questions. Samples coming from different users are randomly interleaved. Figure 3.12 shows the evolution of the F1 score and the number of questions of *Probabilistic context reasoning* with respect to the baselines. The results show that in the early stages, the recognition rate of all considered approaches was not acceptable, highlighting the significance of the active learning module to collect labeled samples for incremental training.

Compared to the *Data-Driven approach*, both the *Deterministic context reasoning* and *Probabilistic context reasoning* approaches quickly achieved high recognition rates and significantly fewer questions. The *Probabilistic context reasoning* approach outperformed the *Deterministic context reasoning* approach, demonstrating a faster learning curve.

However, the number of questions generated by the *Probabilistic context reasoning* approach was only slightly lower than those generated by the deterministic approach, reflecting the results presented in Figure 3.11.

3.7 A System Demonstration

In the previous Sections, we showed that combining context data with common knowledge about the relationship between context and human activities, leads to obtain optimum results in terms of recognition rate by using a small num-



(a) F1 score

(b) percentage of questions

Figure 3.12: Evolution of the recognition model over time. Considered activities: Running, Sitting, Cycling, Standing, Walking, Elevator up, Elevator down, Going Upstairs, Going Downstairs, Brushing Teeth, Moving by car, Sitting transport, Standing transport

ber of labeled training examples. In order to demonstrate to the users how the proposed systems works, we implemented a demo consisting of a real-time activity recognition system that combines supervised learning on inertial sensor data and context-aware reasoning. In particular, we ask the participants to keep a smartphone in a pocket and a smartwatch on the wrist. These devices run custom applications that transmit inertial sensor data and context information to a server that executes our hybrid statistical and deterministic context reasoning approach in real-time. As Figure 3.13 shows, a web dashboard displays in real-time the most important steps executed by our system.

In particular, the dashboard displays to the user the following information:

- In the top-left box, the output of the machine learning module
- In the top-right box, the pre-processed context-data
- In the bottom-left box, the list of consistent and inconsistent activities is derived by context reasoning. By clicking on an activity it is also possible to consult the ontological definition.
- In the bottom-right box, the context-refined probability distribution of activities

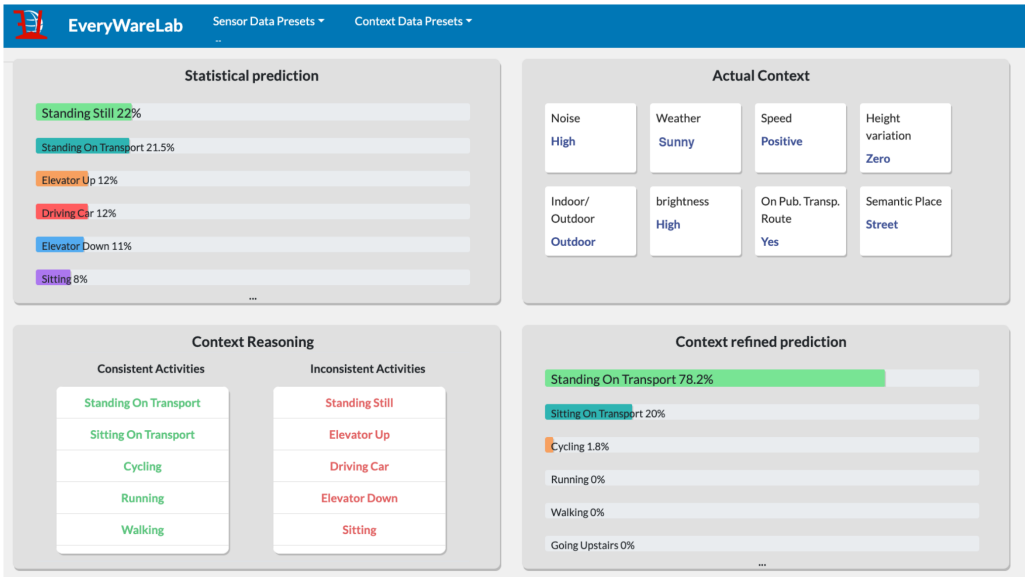


Figure 3.13: Our Dashboard

Given the constraints of the laboratory environment in which the demo took place, we also enabled the users to simulate different contexts and inertial sensors data presets. These presets have been created by using the data collected in the DOMINO dataset, and can be dynamically changed by using a dedicated menu of the dashboard. Example 3.7.1 describes a possible situation in which a user experiences our demo taking advantage of the available data presets.

Example 3.7.1 *Suppose that a user selects from the presets menu the inertial sensor data pattern representing a user standing on a bus. In this case, the classifier shows uncertainty between Standing Still and Standing on Transport activities. This is consistent with theoretical results obtained in Section 3.6. Then, suppose that the user chooses a context preset simulating that she is following a public transportation route and that she is moving at a certain speed. According to the ontology, Standing still is not context-consistent since it should be a static activity. Hence, the system derives Standing on Transport as the most likely context-refined activity and displays it to the user.*

3.8 Summary

In this Chapter, we presented a novel approach that, relying on a combination of semi-supervised learning and knowledge-based reasoning, enables reducing the number of labeled examples required to classify a wide set of physical activities. Overall, the proposed approach uses a machine learning algorithm on inertial sensor data to obtain the candidate probability distribution over the activities carried out by the user. Then, a knowledge-reasoning engine refines the candidate prediction given the relationships between activities and context data modeled into an ontology. Thanks to active learning, our systems can continuously improve the statistical classifier initialized with a limited set of labeled examples. In particular, we developed and evaluated two different ontologies to implement the knowledge-reasoning engine. The former is a deterministic ontology that uses a rigid ontological formalist to model the relationships between activities and context. The latter is a probabilistic ontology that enables capturing more sophisticated probabilistic relationships between activities and context. Our experimental evaluation using the DOMINO dataset showed that the proposed method enables increasing the recognition rate with respect to standard semi-supervised approaches, while reducing the number of active learning questions triggered to the users to obtain annotated training data. In particular, the knowledge-reasoning engine based on probabilistic ontology outperformed the one based on deterministic ontology. Thus, we can conclude that the solution presented in this chapter addresses the research question **Q1)** presented in Section 2.4 by mitigating the data scarcity problem typical of collaborative HAR. We also believe that our approach is more flexible in terms of the availability of context data with respect to using context as features. Since context sources may not always be available, using context as features may lead to missing values in the feature vectors used to update the classifier, which in turn may negatively affect the recognition rate. This work only represents a preliminary investigation of the effectiveness of using probabilistic logics in context-aware and hybrid activity recognition systems. We foresee several promising research directions. First, given the probabilistic ontology, a critical aspect is the setting of weights for the soft axioms determining the influence of context on activities. We plan to

investigate how to populate the probabilistic ontology in a semi-automatic fashion, by extracting knowledge about context and activities from external sources. For instance, some works proposed to extract information from textual description [138] and images [139] of activities from the Web. Those works were mainly focused on building models for smart-home activity recognition. Besides uncertainty on the association of context with activities, a probabilistic ontology may also capture the fact that context data may have an associated confidence value. Indeed, it is not always advisable to completely trust input context data (e.g., geographical positioning, as well as semantic place identification, can have different levels of approximation and reliability). Including uncertainty on input data has the potential of making our system more robust with respect to inaccurate information. Last but not least, it is also important to consider the scalability and privacy issues that may arise in collaboratively training a machine learning model in a centralized way by using the data collected by a wide number of users.

Chapter 4

Federated HAR In Data Scarcity Scenarios

4.1 Introduction

The previous chapter discussed collaborative and semi-supervised methods for mitigating the data scarcity problem in sensor-based HAR [13]. Combining these methods with knowledge-based reasoning can limit user interactions while recognizing a wide range of activities [29, 30]. However, challenges such as scalability and privacy concerns limit the deployment of collaborative approaches. To address these problems, Federated Learning (FL) is introduced as a promising solution for making activity recognition scalable for a large number of users while preserving privacy [19].

Indeed, in traditional collaborative approaches, all data must be transferred to the central server where resides the machine learning model, which can be time-consuming, resource-intensive, and pose privacy risks. Furthermore, as the volume of data increases, it becomes increasingly challenging to store and process it on a single machine. In contrast, federated learning allows each node to train a local model using its own labeled data, forwarding only updated model parameters to the server responsible for aggregating them into a global recognition model. In such a way it is possible to dramatically reduce the amount of data transferred from clients and the server, and decrease privacy risks for the users.

Moreover, the federated approach enables the use of distributed computing resources across many devices, allowing for the efficient processing of large amounts of data. While FL has been successfully applied to HAR, existing FL-based solutions assume complete availability of labeled data at each node, which is not realistic for HAR applications with limited labeled data availability [23, 98]. Extending FL to semi-supervised learning is one of the open challenges in this area [98].

In this Chapter, we propose FedAR: a hybrid semi-supervised and FL framework that enables privacy-aware and scalable HAR based on mobile and wearable devices. Different from the majority of the existing solutions, FedAR considers a limited availability of labeled data. In particular, FedAR combines active learning and label propagation to provide labels to a large amount of unlabeled data. Newly labeled data are periodically used by each node to perform local training, thus obtaining the model parameters that are then transmitted to the server that aggregates them using Secure Multiparty Computation. FedAR also relies on transfer learning to fine-tune the global model for each user, while generating a global model that generalizes over unseen users. Given the limitations of existing evaluation methodologies for FL applied to HAR [20], we also designed a novel evaluation methodology to robustly assess both the generalization and the personalization capabilities of our approach. The results obtained with two publicly available datasets showed that FedAR reaches a recognition rate close to state-of-the-art solutions that assume the complete availability of labeled data. Moreover, the number of generated active learning questions resulted very small and hence acceptable for real-world deployment.

This chapter is structured as follows. The proposed methodology and the related algorithms of FedAR are detailed in Section 4.2. We introduce the experimental evaluation in Section 4.3. Finally, in Section 4.4 we summarize our contributions and discuss the actual advantages and limitations.

4.2 The Proposed Methodology

4.2.1 System Architecture

The overall architecture of the proposed approach, is depicted in Figure 4.1. In

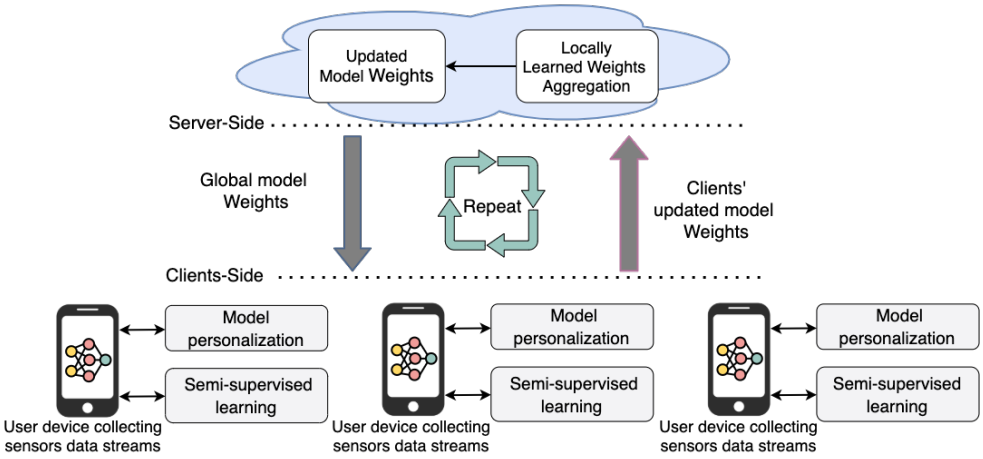


Figure 4.1: Overall architecture of the proposed approach.

particular, by following the FL framework, the actors of FedAR are a server and a set of clients that cooperate to periodically compute the weights of a global activity recognition model. In order to address the labeled data scarcity problem, the proposed approach initializes the global model in an offline phase with a limited amount of labeled data, while each client implements a semi-supervised learning strategy (i.e., a combination of active learning and label propagation) to semi-automatically label a portion of the unlabeled sensor data stream. Periodically (e.g., every night), the server starts a process to update the weights of the global model. Each client uses its available labeled data to train its local model. The resulting local weights are transmitted to the server, which aggregates them with the ones from the other clients to obtain a new version of the global model. Finally, the new version of the global model weights is transmitted to each client.

Since different users may perform activities in very different ways, a model personalization module on each client is in charge of fine-tuning the updated global model weights on the specific user. A more detailed overview of the global model

update and personalization is described in Section 4.2.4.

4.2.2 Local Models

One of the strengths of the proposed approach is that it considers both personalization and generalization aspects. Personalization is crucial for the local models to recognize the activities of each user more accurately. On the other hand, generalization is a desirable property for the global model. Indeed, some participating users may not wish to collect labeled data (not even a small amount) or may have devices not adequate to perform local training. Those users are not able to actively contribute to the federated learning process, and their clients would directly use the last version of the global model for activity classification.
In order to guarantee both personalization and generalization, each client stores two distinct instances of the activity model. The former is called *Shareable Model*, and it is the one used for federated learning. In order to personalize the activity model for each user, a straightforward solution would be to fine-tune the *Shareable Model* with transfer learning approaches [140]. However, recent studies show that a global model built by aggregating the weights of fine-tuned models exhibits poor generalization capabilities on external users [20]. In order to overcome this problem, at the end of each global model update the clients that actively contribute to the federated learning process create a copy of the *Shareable Model* that is called *Personalized Model*. The *Personalized Model* is fine-tuned on the specific user and it is used for activity classification. Besides improving generalization, an advantage of keeping private the weights of the *Personalized Local Model* is a positive impact on privacy protection [141].

4.2.3 Semi-supervised Data Labeling And Classification

Figure 4.2 depicts the semi-supervised data labeling and classification flow of the proposed approach. Each client in FedAR uses the *Personalized Model* to classify activities in real-time on the continuous stream of unlabeled pre-processed sensor data. Before classification, each unlabeled data sample is stored in the *Feature Vectors Storage*. This storage collects both unlabeled and labeled data samples. After classification, if the confidence over the current prediction is below

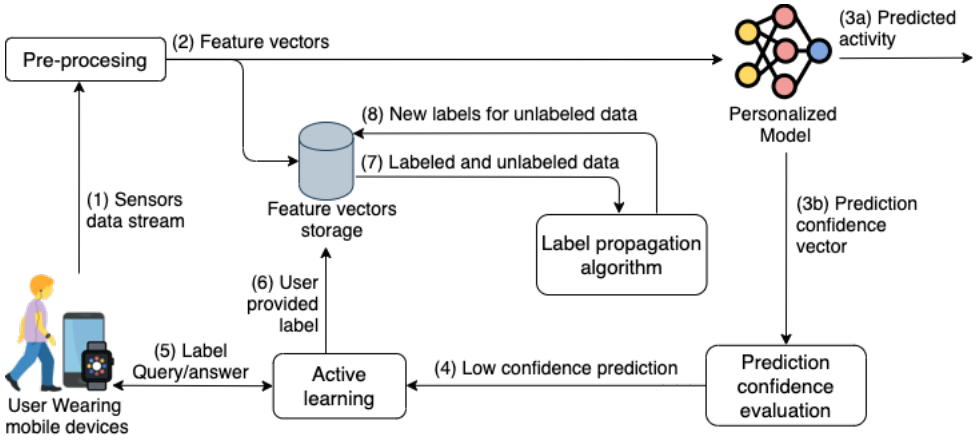


Figure 4.2: Semi-supervised data labeling and classification data flow

a threshold, an active learning process is started, and the system asks the user about the activity that she was actually performing. The feedback from the user is then associated with the corresponding feature vector in the *Feature Vectors Storage*. Active learning makes it possible to assign a label to those informative data points that can effectively improve the local model. For the sake of usability, the number of active learning queries should be low, since they may bother the user during activity execution. For this reason, FedAR also periodically applies a Label Propagation algorithm to spread the labels acquired through active learning to a larger number of unlabeled data points. The advantage of label propagation is to further improve the recognition rate by training the classifier with a significant amount of labeled data samples and, at the same time, to reduce the number of needed active learning queries over time.

4.2.4 Global Model Update And Personalizazion

Periodically (e.g., every night) the server asks to the participating clients to update the global model. This process is depicted in Figure 4.3. First, each client replaces its *Shareable Model* with the current version of the *Global Model*. Then, the labeled data in the *Feature Vectors Storage* are used to perform local training of the *Shareable Model*. After training, the updated *Shareable Model* weights are then forwarded to the server, that is in charge of aggregating the

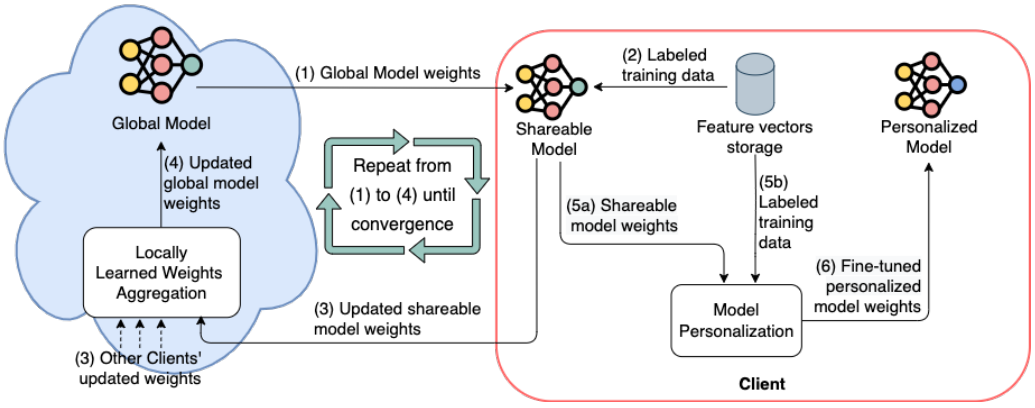


Figure 4.3: Local models training and personalized model update

weights from all the clients to generate a new version of the global model. These steps are repeated until the convergence of the global model. At the end of this process, the *Shareable Model* of each client is replaced with the last stable version of the *Global Model*. Then, the *Model Personalization* module generates a copy of the *Shareable Model* that is called *Personalized Model*, that is fine-tuned using the *Feature Vectors Storage*. The result of this process is a *Personalized Model* that takes advantage of the high-level features of the *Global Model* as well as the personalized aspects of the specific user.

In this section, we describe in detail the algorithms of FedAR.

4.2.5 The Activity Model

Since we consider a setting with limited availability of labeled data, activity models that automatically learn features from raw data are not effective in FedAR. Indeed, based on our experiments that we describe in Section 4.3.2, CNN models reach significantly lower recognition rates in FedAR due to the high complexity of learning reliable features from limited labeled data. For this reason, in FedAR, the activity classification model is based on a fully-connected deep learning model, and the input is a vector of handcrafted features. In particular, we choose features that proved to be effective for HAR [29]. Recent studies in the HAR domain demonstrate that a good choice of handcrafted features and fully connected models can lead to recognition rates comparable to the ones of state-

of-the-art CNN models [142].

4.2.6 Initialization Of The Global Model

At the very beginning, the participating clients in FedAR need a pre-trained global model to infer labels on unlabeled data. However, in this work we assume limited availability of labeled data.

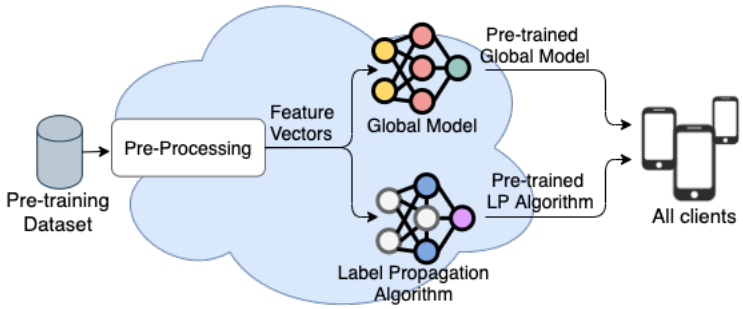


Figure 4.4: Initialization of the global model in FedAR.

Hence, FedAR initializes the global model using a restricted annotated dataset (we will call it *pre-training dataset* in the following) [1]. The *pre-training dataset* is also used to initialize label propagation algorithm. In realistic settings, the *pre-training dataset* can be, for example, a combination of publicly available datasets, or a small training set specifically collected by a restricted number of volunteers. Figure 4.4 summarizes the initialization mechanism of FedAR.

4.2.7 The Proposed Federated Learning Based Approach

In the following, we describe the FL process to update the global and local models. Periodically (e.g., each night) the server starts a global model update process. The devices that are available to perform computation (e.g, the ones idle and charging) inform the server that they are eligible to take part in the FL

[1]Note that, considering out target application, a labeled dataset is a collection of timestamped inertial sensors data acquired from mobile/wearable devices during activity execution. Examples of such sensors are accelerometer, gyroscope, and magnetometer. The labels are annotated time intervals that indicate the time-span of each performed activity.

process. Afterward, the server executes several communication rounds to update the weights of the global model.

A *communication round* consists of the following steps:

- The server sends the latest version of the global weights to a fraction of the eligible devices
- Each device uses the labeled data in the *Feature Vectors Storage* to train the *Shareable Model*
- When local training is completed, each device sends the new weights of the *Shareable Model* to the server
- The server aggregates the local weights to compute the new global weights

The communication rounds are repeated until the global model converges. In particular, the server considers the global model converged when there is no substantial difference between the weights of the global model after a certain number of subsequent updates [2]. Then, the new weights are transmitted to each participating device including the ones that did not actively contribute to the communication rounds. The server updates the global model weights by executing a weighted average of the locally learned model weights provided by clients. Since the local weights may reveal private information, the aggregation is performed using the Secure Multiparty Computation approach presented in [97]. The pseudo-code of the server-side federated learning process is described in Algorithm 1, while the client-side in Algorithm 2.

4.2.8 Model Adaptation

FedAR adopts a transfer learning inspired strategy to fine-tune the *Personalized Model* on each user. The intuition behind the user adaptation mechanism is that the last layers of the neural network (i.e., the ones closer to the output) encode personal characteristics of activity execution, while the remaining layers encode more general features that are common between different users [107]. As

[2] In Section 4.3.3 we discuss pro and cons of the criteria that are generally used to evaluate the convergence of a federated learning model in HAR.

Algorithm 1 Server side - Federated global model

$PT \Leftarrow$ pre-training set
Initialize global model w^G with PT
$d \Leftarrow$ participating devices
for each periodic update (e.g., every night) **do**
    for each communication round **do**
        ask for eligibility to each device in d
        $ed \Leftarrow$ eligible devices
        $ed' \Leftarrow k$ devices randomly sampled from ed
        send w^G to each device in ed'
        aggregate updated models' weights received from devices in ed' with SMC [97]
    end for
end for

Algorithm 2 FedAR - Client side - Model update

$pm \Leftarrow$ Personalized Model
$sm \Leftarrow$ Shareable Model
Update the *Feature Vectors Storage* using the Label Propagation algorithm in Section 4.2.8.
for each communication round i **do**
    train sm using labeled data in the Feature Vector Storage
    send sm to the server
    receive updated global model w_i^G
    $sm \Leftarrow w_i^G$
end for
$pm \Leftarrow sm$
fine-tune pm using the transfer learning inspired method described in Section 4.2.8

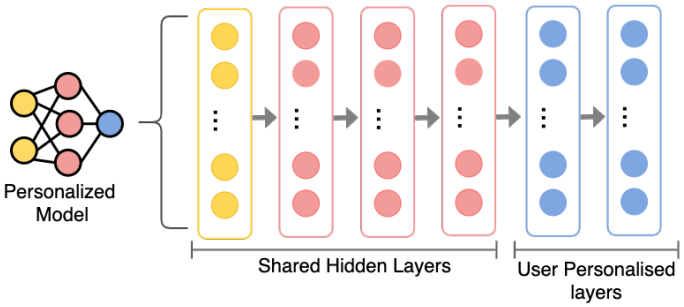


Figure 4.5: Shared and Personal Layers.

depicted in Figure 4.5, we refer to the last l layers of the neural network as the *User Personalized Layers*, while we refer to the remaining ones as *Shared Hidden Layers*. In FedAR, when the update of the global model is complete, each client creates the *Personalized Model* as a copy of the *Shareable Model*. In order to fine-tune the *Personalized Model* on each user, the *Shared Hidden Layers* are frozen, and the *Feature Vector Storage* is used to train the *User Personalized Layers*.

Active Learning and Labels propagation

In the following, we describe how each client semi-automatically provides labels to the stream of unlabeled sensor data. FedAR relies on a combination of two semi-supervised learning techniques: *Active Learning* and *Label Propagation*.

Active Learning

An active learning process requires the user feedback about her currently performed activity when there is uncertainty in the classifier's prediction. The intuition is the following: unlabeled data samples for which the classification confidence is significantly low would have the most impact in improving the classifier if the label were available (i.e., they are the more informative ones). FedAR relies on the active learning strategy previously presented in Section 3.5 of Chapter 3. Algorithm 3 introduces the pseudo-code describing how FedAR executes classification and active learning.

Algorithm 3 Client side - Classification and data labeling

1: $sm \Leftarrow$ Shareable Model
2: $pm \Leftarrow$ Personalized Model
3: receive pre-trained w^G from the server
4: $sm \Leftarrow w^G$
5: $pm \Leftarrow w^G$
6: **for** each feature vector fv computed in real-time from sensor data **do**
7: $\vec{p} \Leftarrow$ probability distribution of the activities predicted by pm on fv
8: output the most likely activity according to $\vec{p}$
9: **if** a feedback is needed according to VAR-UNCERTAINTY [132] **then**
10: $l \Leftarrow$ activity label from the user
11: add (fv, l) to the Feature Vectors Storage
12: **else**
13: add $(fv, -)$ to the Feature Vectors Storage ▷ unlabeled data point
14: **end if**
15: **end for**

Label Propagation

The major drawback of active learning is that the queries may interrupt the user while performing an activity. In order to reduce the interaction with the user and, at the same time, to train the local models with a larger amount of labeled data, FedAR also relies on label propagation.

The Label Propagation process is started when the server requires to update the global model (see Algorithm 1). Given a set of labeled and unlabeled data points, the goal of label propagation is to automatically spread labels to a portion of unlabeled data [143]. The intuition behind label propagation is that data points close in the feature space likely correspond to the same class label. The Label Propagation model of FedAR is a fully connected graph $g = (V, E)$ where the nodes V are all the data samples in the *Feature Vectors Storage* and the weight on each edge in E is the similarity between the connected data points. In the literature, this similarity is usually computed using K-Nearest Neighbors (KNN) or Radial Basis Function Kernel (RBF kernel). FedAR relies on the RBF kernel due to its trade-off between computational costs and accuracy [144]. Formally, the RBF kernel function is defined as $K(x, x') = e^{-\gamma||x-x'||^2}$ where $||x - x'||^2$ is the squared Euclidean distance between the feature vectors of two nodes x and

x' (where x' is a labeled node), and $\gamma \in \mathbb{R}+$. Hence, the value of the RBF kernel function increases as the distance between data points decreases. The kernel is used to perform inductive inference to predict the labels on unlabeled data points, based on a threshold on the similarity between the nodes. This process is repeated until convergence (i.e., when there are no more unlabeled data points for which label propagation is reliable based on the threshold). In FedAR, the Label Propagation model (i.e., the graph) is initialized with the labeled data points of the *pre-training dataset*. Moreover, this model is personal and never shared with other users nor with the server.

4.3 Experimental Evaluation

In the following, we describe the methodology designed to evaluate the effectiveness of FedAR, both in terms of personalization and generalization. Finally, we discuss the obtained results. Since FL makes sense when many users participate in collaboratively training the global model, we considered publicly available datasets of physical activities (performed both in outdoor and indoor environments) that were collected involving a significant number of subjects. However, there are only a few public datasets with these characteristics. Two of them are the MobiAct [121] and the WISDM [51] datasets, which we detailed in Section 2.3. In particular, MobiAct includes labeled data from 60 different subjects with high variance in age and physical characteristics. The dataset contains data from inertial sensors (i.e., accelerometer, gyroscope, and magnetometer) of a smartphone positioned in a trousers' pocket freely chosen by the subject in any random orientation. In our experiments, we take into account only the following physical activities: *standing*, *walking*, *jogging*, *jumping*, and *sitting*, omitting those with a limited number of samples. Indeed, our evaluation methodology requires partitioning the data of each user and those activities with a small number of samples would be insufficiently represented in each partition. We believe that this problem is only related to this specific dataset and that, in realistic settings, even short activities would be represented by a sufficient number of samples.

Given the WISDM dataset, it contains accelerometer data collected from a smartphone located in the front pants leg pocket of each subject during activity

execution. WISDM includes data from 36 subjects. The activities included in this dataset are the following: *walking*, *jogging*, *sitting*, *standing*, and *taking stairs* (and we consider all of them).

4.3.1 A Novel Evaluation Methodology

We split each of the considered datasets into three partitions that we call Pt, Tr, and Ts. The partition Pt (i.e., pre-training data) contains data of users that we only use to initialize the global model. Tr (i.e., training data) is the dataset partition that includes data of users who participate in FL. Finally, Ts (i.e., test data) is a dataset partition that includes data of left-out users that we only consider to periodically evaluate the generalization capabilities of the global model. In our experiments, we randomly partition the users as follows: 15% whose data will populate Pt, 65% whose data will populate Tr, and 20% whose data will populate Ts. We partition the data for each user in Tr into sh shards of equal size. In realistic scenarios, each shard should contain data collected during a relatively long time period (e.g., a day) where a user executes many different activities. However, the considered datasets only have a limited amount of data for each user (usually less than one hour of activities for each user). Hence, we generate shards as follows. Given a user $u \in Tr$, we randomly assign to each shard a fraction $\frac{1}{sh}$ of the available data samples associated with u in the dataset. Note that each data sample of a user is associated with exactly one shard. This mechanism allows us to simulate the realistic scenario described before, where users perform several types of activities in each shard.

Evaluation Algorithm

In the following, we describe our novel evaluation methodology step by step. First, the labeled data in Pt are used to initialize the global model, which is then distributed to the devices of all the users in Tr that will use it as the first version of the *Personalized Model*. We evaluate the recognition capabilities of the initial pre-trained global model on the partition Ts in terms of the F1 score. This assessment allows us to measure how the initial global model generalizes on

unseen users before any FL step.

As we previously mentioned, for each user, we partition its data samples in Tr into exactly sh shards. For the sake of the evaluation, we assume a synchronous system in which the shards of the different users in Tr are actually temporally aligned and occur simultaneously (i.e, the first shards of every user occur at the same time interval, the second shards of every user occur at the same time interval, and so on). Note that, in the considered datasets, each user has a different data distribution and a different number of samples. Hence, within a specific shard, each client contributes with data collected considering its personal distribution. The evaluation process is composed of sh iterations, one for each shard. Considering the i-th shard we proceed as follows:

1. The devices of the users in Tr exploits the *Personalized Model* to classify the continuous stream of inertial sensor data in its shard. We use the classification output to evaluate the recognition rate in terms of F1 score providing an assessment of personalization. Note that, during this phase, we also apply our active learning strategy and we keep track of the number of triggered questions.

2. When all data in the shard have been processed (by all devices), the server starts a number r of communication rounds with a subset of the devices in order to update the global weights. Each round is implemented as follows:

   (a) The server randomly selects a certain percentage $p\%$ of users in Tr and sends to their devices the last update of the global weights.

   (b) Each user's device, by receiving the global weights, applies Label Propagation (See Section 4.2.8) and uses the newly labeled data to train its *Shareable Model*. After training, the resulting weights are transmitted to the server.

   (c) The server merges the received weights obtaining a new version of the global model weights.

   (d) We evaluate in terms of F1 score the recognition rate of the resulting global model on the left-out users in Ts (providing an assessment of

generalization).

3. After the execution of all the communication rounds, each users' device:

   (a) replaces the weights of the *Shareable Model* and *Personalized Model* with the ones of the latest global model

   (b) fine-tunes the *Personalized Model* with labeled data from active learning and label propagation

   (c) starts the personalization process described in Section 4.2.8.

Note that our evaluation methodology introduces several levels of randomness: assigning users to Ts, Tr, and Pt; assigning data samples to shards; selecting devices at each communication round. We iterate experiments 10 times and average the results in order to make our estimates more robust.

4.3.2 Results

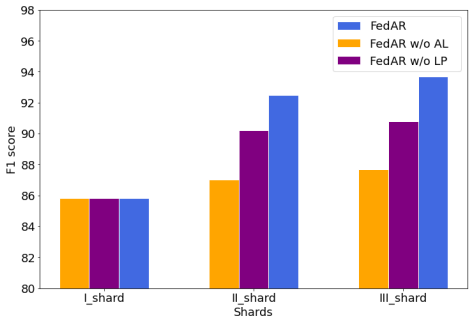
In the following, we report the results of the evaluation of FedAR.

Classification model and hyper-parameters

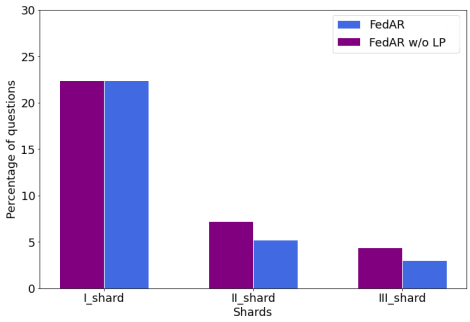
As explained and motivated in Section 4.2.5, our classification model is a fully-connected deep neural network. The network consists of four fully connected layers having respectively 128, 64, 32, and 16 neurons, and a softmax layer for classification. We use Adam [145] as optimizer. The choice of this specific network architecture is due to the good performance reported in the federated HAR literature [103]. As hyper-parameters, we empirically chose $w = 4s$, $p = 30\%$, $r = 10$, $l = 2$, $sh = 3$, and 10 local training epochs with a batch size of 30 samples. These hyper-parameters have been empirically determined based on data in Ts. The low number of epochs and communication rounds is due to the small size of the public datasets. This also limits the data in each shard. In a large-scale deployment, these parameters should be accurately calibrated.

Impact Of Semi-Supervised Learning

Figure 4.6 and Figure 4.7 show how the F1 score and the percentage of active learning questions change at each shard for the users in Tr.

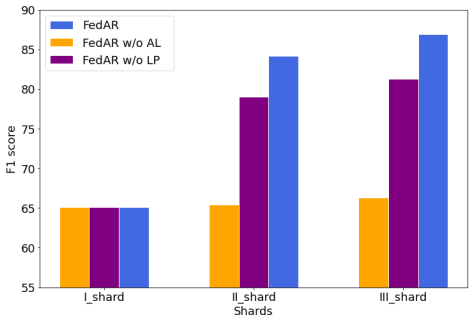


(a) Average F1 score

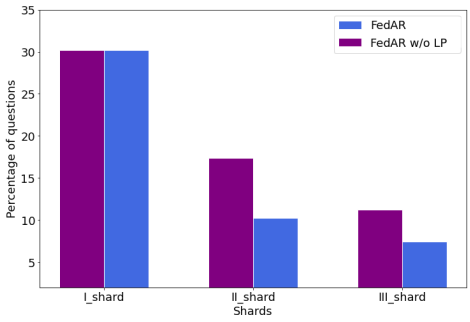


(b) Average percentage of active learning questions.

Figure 4.6: MobiAct: The impact of label propagation and active learning on the subjects that participated in the FL process.



(a) Average F1 score



(b) Average percentage of active learning questions.

Figure 4.7: WISDM: The impact of label propagation (LP) and active learning (AL) on the subjects that participated in the FL process.

We observe that the F1 score significantly improves shard after shard, while the number of active learning questions decreases. Averaging the results of both datasets, the number of active learning questions at the first shard is around 25%, while at the last shard is only around 5%. This result indicates that our

method continuously improves the recognition rate with a limited amount of labels provided by the users. Moreover, the continuous decrease in the number of questions militates for the usability of our method, which will prompt fewer and fewer questions with time. These figures also show the impact of combining active learning with label propagation. Without label propagation, active learning alone leads to a lower recognition rate and a higher number of questions. This means that the labeled data points derived by label propagation positively improve the activity model. On the other hand, we observe that label propagation leads to unsatisfying results without active learning. Indeed, the labeled samples obtained by active learning represent informative data that are crucial for label propagation. Hence, the evaluation with both datasets confirms that the combination of active learning and label propagation leads to the best results.

In Figure 4.8 and Figure 4.9 we show the generalization capability of the global model on left-out users (i.e., users in partition Ts) after each communication round performed during the FL process with the users in Tr. The red lines

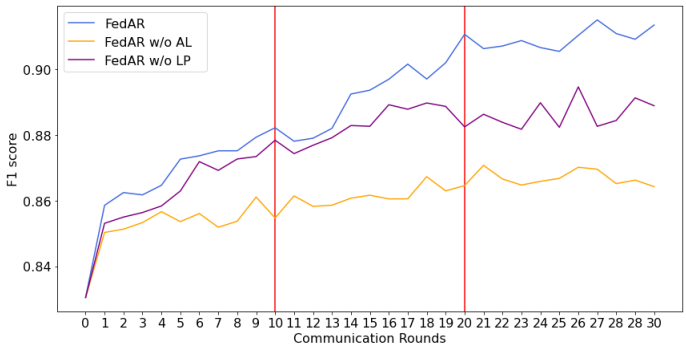


Figure 4.8: MobiAct: the trend of F1 score on the left-out users after each communication round. This Figure also shows the impact of active learning and label propagation. Each red line marks the end of a shard.

mark the end of each shard. The results indicate that the federated model constantly improves also for those users that did not contribute with training data, even if the active learning questions continuously decrease. These plots also confirm that the combination of label propagation and active learning leads to the best results on both datasets.

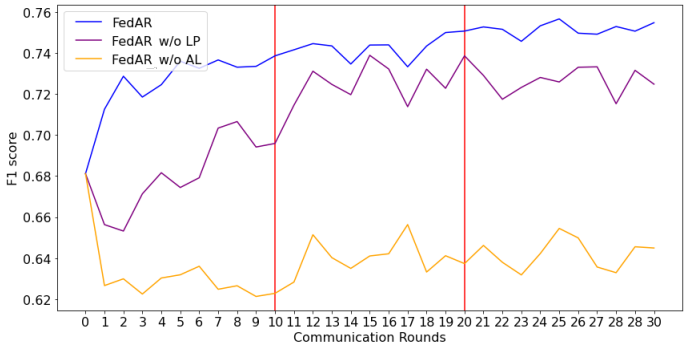
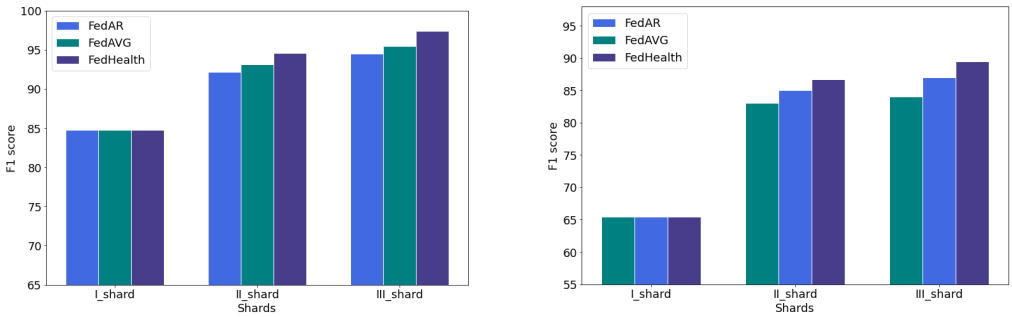


Figure 4.9: WISDM: the trend of F1 score on the left-out users after each communication round. This Figure also shows the impact of active learning and label propagation. Each red line marks the end of a shard.

The Proposed Approach Vs Fully Supervised Solutions

We compared our approach with two existing FL methods based on fully labeled data. The first one is the well-known FedAVG [19], which is the most common FL method in the literature. FedAVG simply averages the model parameters derived by the local training on the participating nodes (without any personalization). The second method that we use for comparison is called FedHealth [23]. This is one of the first FL approaches proposed for activity recognition on wearable sensor data. Similarly to our approach, FedHealth applies personalization using a transfer learning based strategy. Since FedAR considers a limited amount of available labeled data, our goal is to achieve a recognition rate that is as close as possible to the one obtained by solutions that assume full availability of annotations. For the sake of fairness, in our experiments we adapted FedAVG and FedHealth to use the same neural network that we use in FedAR. Hence, we performed our experiments using our evaluation methodology by simulating that, for FedAVG and FedHealth, each node has the ground truth for each data sample on each shard. Hence, the evaluation of those methods does not include active learning and label propagation. Moreover, differently from FedAR, FedAVG and FedHealth only use a single local model.

The results of this comparison for the users in Tr (i.e., the ones that actively participated in the FL process) are reported in Figure 4.10a and Figure 4.10b.



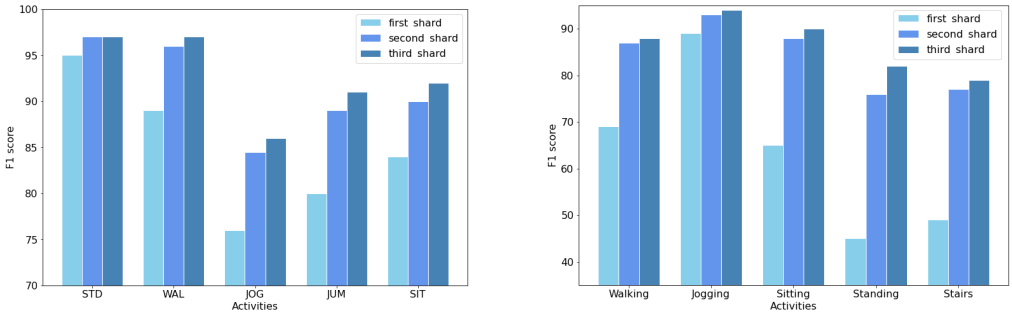
(a) MobiAct (b) WISDM

Figure 4.10: Comparison of FedAR with methods based on fully labeled data.

From these plots, we observe that FedAR reaches recognition rates that are similar to solutions based on fully labeled data at each shard. The advantage of FedAR is that it can be used for realistic HAR deployments where the availability of labeled data is scarce. Despite a reduced number of required annotations, FedAR performs even better than FedAVG on the WISDM dataset, while on MobiAct it performs slightly worse. Moreover, FedAR is only $\approx 3\%$ behind FedHealth on both datasets.

Performance on each activity

Figure 4.11 shows how the recognition rate improves between shards for each activity for the users in Tr on both datasets. We observed an improvement in



(a) MobiAct (b) WISDM

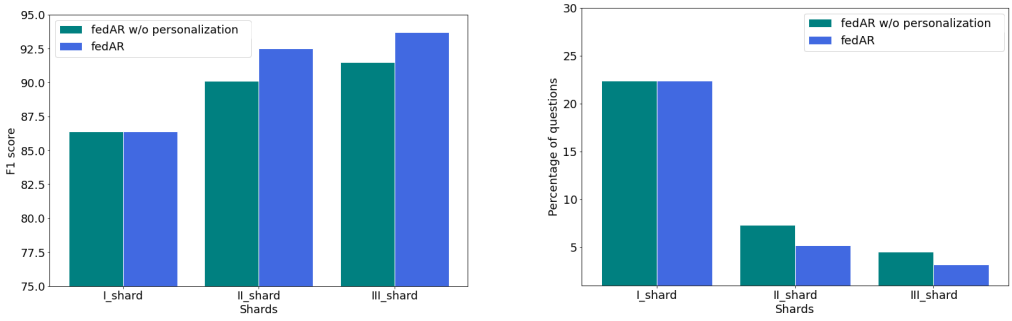
Figure 4.11: F1 score at each shard for each activity on the users that participated in the FL process.

the recognition rate shard after shard for each considered activity. The only exception is the *standing* activity on the MobiAct dataset in the third shard, which maintains the same F1 score.

Overall, the greatest improvement occurs between the first and the second shards. This is due to the fact that, in the first shard, activities are recognized using the initial global model only trained with the *pre-training dataset*. Starting from the second shard, classification is performed with the *Personalized Model* updated thanks to FL and personalized using our transfer learning based approach.

Impact of personalization

Figure 4.12 and Figure 4.13 show the impact of the FedAR personalization strategy based on transfer learning. This evaluation is performed on the users in the Tr partition.

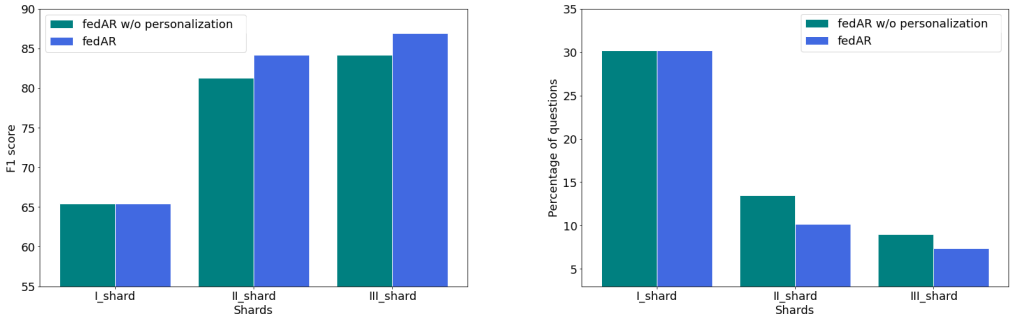


(a) Average F1 score

(b) Average percentage of active learning questions

Figure 4.12: MobiAct: results on the users that participated in the FL process for each shard, with and without personalization.

As expected, fine-tuning the personal models leads to an improvement both in the recognition rate and in the number of questions in active learning. Note that, during the first shard, classification is performed using the weights derived from the *pre-trained dataset* and personalization is applied since the second shard.



(a) Average F1 score

(b) Average percentage of active learning questions

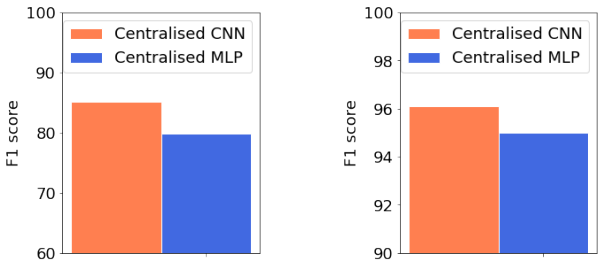
Figure 4.13: WISDM: results on the users that participated in the FL process for each shard, with and without personalization.

Fully Connected vs Convolutional models

The classification model in FedAR is a fully connected network (we will refer to it as MLP[3] for the sake of brevity) that receives as input handcrafted feature vectors. Nonetheless, as we introduced in Section 2.1.1, in the literature, Convolutional Neural Networks (CNNs) proved to be very effective in fully supervised HAR approaches, since they can automatically learn features from raw data [142].

We performed a preliminary experiment to compare MLP and CNN in a fully supervised centralized approach using a leave-one-subject-out cross-validation. As CNN architecture, we consider the one recently proposed in [146] since it proved to be one of the most effective for sensor-based HAR. Figure 4.14 shows the outcome of this comparison. We observe that considering a fully-supervised centralized setting, CNN is more effective on both datasets. However, we observed that CNN struggles in learning reliable features considering our federated and semi-supervised setting, since the amount of labeled data to train the classifier is limited (cold start issue). Figures 4.15 and 4.16 show the comparison of FedAR using our MLP model with handcrafted features and the CNN model. On both datasets, MLP quickly reaches a higher F1 score with respect to CNN with a significantly lower number of active learning queries. Since features are

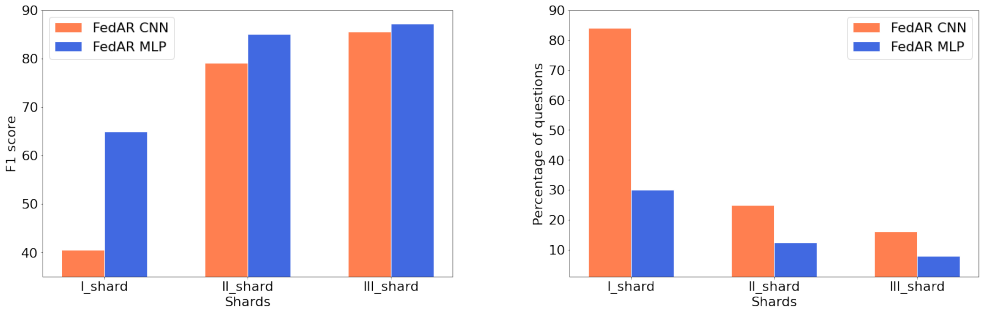
[3]MultiLayer Perceptron



(a) WISDM (b) MobiAct

Figure 4.14: Centralized setting: MLP vs CNN based on leave-one-subject-out cross-validation.

computed a priori, the MLP model can immediately focus on training the classification layers rather than learning features. Hence, these results motivate our choice of adopting a MLP model with handcrafted features in FedAR.



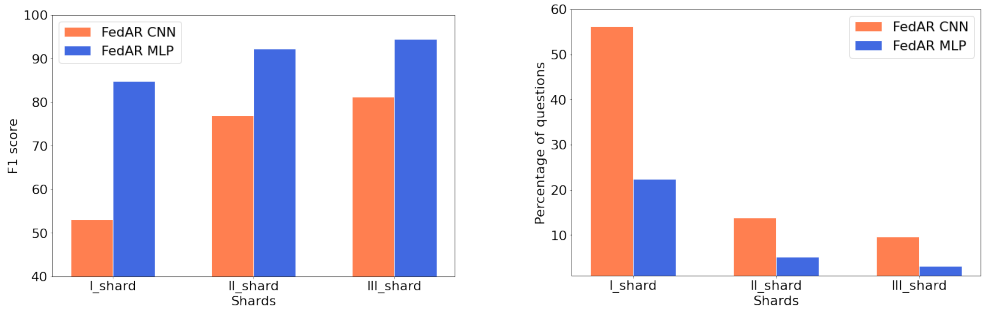
(a) Average F1 score (b) Average percentage of active learning questions

Figure 4.15: WISDM: results on the users that participated in the FL process for each shard using both CNN and MLP networks

4.3.3 Discussion

Generality Of The Proposed Approach

While we designed FedAR with wearable-based activity recognition as target application, we believe that this combination of semi-supervised and FL can be applied also to many other applications. Our method is suitable for human-centered classification tasks that include the following characteristics:



(a) Average F1 score

(b) Average percentage of active learning questions

Figure 4.16: MobiAct: results on the users that participated to the FL process for each shard using both CNN and MLP networks

- There is a large number of clients that participate in the FL process.
- Classification needs to be performed on a continuous data stream, where labels are not naturally available.
- Each node generates a significant amount of unlabeled data.
- It is possible to periodically obtain the ground truth by delivering active learning questions to users that are available to provide a small number of labels.
- It is possible to obtain a limited training set to initialize the global model. Hence, a small group of volunteers should be available (in an initial phase) for annotated data acquisition.
- The nodes should be capable of computing training operations. Clearly, nodes can also rely on trusted edge gateways/servers (like proposed in [103]).

Convergence of the global model in Federated HAR

Evaluating the convergence of the global model is essential for determining the quality of the global model and ensuring its effectiveness in real-world applications. Here, we discuss the most common approaches and corresponding challenges associated with convergence evaluation in federated learning.

There are three primary approaches commonly used to evaluate convergence in federated learning:

1. Server-side evaluation: A portion of the data is held out on the server side and used as an evaluation set for the global model. The global model is trained using data from multiple clients, and its performance on the evaluation set is monitored to evaluate convergence. The global model is considered converged when its performance on the evaluation set reaches a satisfactory level. Server-side evaluation is computationally efficient but may not reflect the performance of the model on all clients' data.

2. Client-side evaluation: In this approach, each client holds out a portion of its data and uses it as an evaluation set. The client trains a local model using its data and sends the updated model parameters to the server. The server evaluates the performance of the global model on the held-out data from each client and monitors the convergence of the model across clients. When the model is converged for most of the clients, it can be considered converged. Client-side evaluation provides a more comprehensive evaluation of the model's performance, but it can be more computationally expensive.

3. Parameter similarity: The server considers the global model converged when there is no substantial difference between the parameters or gradients of the global model after a certain number of subsequent updates. The server can use a threshold to determine the level of similarity required for convergence. Parameter similarity is a simple and efficient approach, but it may not guarantee optimal performance on all clients' data.

Each approach has its own advantages and limitations, and the choice of approach depends on the specific application requirements and available resources. Moreover, evaluating convergence can be challenging in federated learning due to the heterogeneity of the data across clients (i.e., non-iid data), and the presence of communication delays and failures.

In this thesis we adopted the Parameter similarity criteria to determine the convergence of the global model due to its simplicity and efficiency. However, given the importance of convergence evaluation and the many challenges that

emerges in this area, we can consider it as an opened research problem that we aim to study in deep in the near future.

4.4 Summary

In this Chapter, we presented FedAR, a novel semi-supervised federated learning framework for activity recognition on mobile devices. This approach addresses the research question **Q2)** presented in Section 2.4 by providing a semi-supervised and collaborative learning solution that enables a privacy-aware and scalable activity recognition, considering also the data scarcity problem. To the best of our knowledge, FedAR is the first application of federated learning to personalized activity recognition that is not based on the assumption that labeled data exists for all participating clients. Our results showed that the combination of active learning and label propagation leads to recognition rates that are comparable to the ones reached by solutions that rely on fully supervised learning to train the local models. Moreover, the personalization strategy implemented by FedAR enables fairly mitigating non-IID problem typical of FL-based approaches for HAR. By following this promising research direction, in the next Chapter, we will introduce a federated clustering method that allows further reducing the non-IID problem for FL-based HAR. Indeed, it has been shown that HAR is more effective when the collaborative model only involves users that are similar between them. [134].

Chapter 5

Cluster-based And Semi-Supervised FL for HAR

5.1 Introduction

As we introduced in the previous Chapters, in the last few years the Federated Learning paradigm attracted attention from the HAR community, as an enabling technology to mitigate scalability and privacy problems related to collaborative learning solutions for human activity recognition [20, 22, 23]. Even though FL is a promising direction toward real-world HAR, there are still some limitations. A major issue is that the FL global model should generalize over a large number of users. However, different users may perform activities with distinctive patterns depending on their physical characteristics, age, and habits. Indeed, data coming from different users likely results non-independently and identically distributed (non-IID) [106].

In Chapter 4, we presented *FedAR*, a semi-supervised approach for Federated Learning HAR that uses a fine fine-tuning strategy based on transfer learning in order to mitigate this issue. However, this approach struggles to balance personalization and generalization in large-scale scenarios. In the general literature on FL, *Federated Clustering* has been recently proposed to address the non-IID problem [109, 111]. Nonetheless these solutions do not consider the data scarcity

issues typical in the HAR domain.

In this Chapter, we propose SS-FedCLAR: a novel Semi-Supervised Federated Clustering method for Personalized Sensor-Based Human Activity Recognition. With respect to existing federated clustering approaches, SS-FedCLAR selects only a portion of the model weights shared by each client, with the objective of computing a similarity score and building groups of users using a hierarchical clustering algorithm. The selected weights intuitively characterize the subject-specific activity patterns. For instance, considering deep learning models, these would be the weights corresponding to layers that are closer to the output [107]. In SS-FedCLAR, those users that can not be included in any cluster will use a generic global model that is trained by all the participating users, like in a standard federated learning setting. Moreover, a transfer learning based method is used to fine-tune activity recognition on each user to further improve personalization. Finally, in order to deal with the data scarcity issue, SS-FedCLAR also implements the hybrid active learning and label propagation strategy that we proposed in Section 4.2.8.
We evaluated SS-FedCLAR on two well-known public datasets of sensor-based HAR and our results show that FedCLAR outperforms FL-based state-of-the-art semi-supervised approaches for HAR that use transfer learning to tackle the non-IID problem. Furthermore, our experimental evaluation also shows the advantage of combining federated clustering with a fine-tuning strategy inspired by transfer learning to improve personalization.

The rest of the Chapter is structured as follows. In Section 5.2 we formalize the problem of non-IID data in the field of FL-based HAR. The proposed methodology and algorithms are hence described in Section 5.3. Then, Section 5.4 introduces the experimental evaluation and the obtained results. Finally, Section 5.5 summarizes our contributions and concludes the Chapter.

5.2 Non-IID Issue in HAR

Let $U = \{U_1, \ldots, U_n\}$ be the set of users. Each user U_i is associated with a labeled dataset $D_i = \{(x, y)\}$, where x is a data point and y the corresponding activity label. Let $D = \{D_1, \ldots, D_n\}$ be the set of datasets, each one corresponding to a user in U. D is non-independently and identically distributed (non-IID) if at least a pair of datasets $D_i, D_j \in D$ satisfies one of the following conditions [109]:

- **Feature distribution skew**: $P_{D_i}(x) \neq P_{D_j}(x)$. This inequality between probability distributions is true when the data samples in D_i have a significantly different marginal distribution than the ones in D_j. In HAR, this often happens since each subject may perform activities in a peculiar way. Among many factors, users' physical characteristics have a strong impact on activity patterns. For instance, a young subject would probably have a faster walking pattern than an elder subject.

- **Label distribution skew**: $P_{D_i}(y) \neq P_{D_i}(y)$. This inequality between probability distributions is true when the labels in D_i have a significantly different marginal distribution than the ones in D_j. In HAR, this usually happens since different users may have different daily routines. For example, a sporty subject would likely spend more time *running* or *cycling* than a sedentary subject.

- **Quantity distribution skew**: This condition is true when $|D_i|$ and $|D_j|$ are significantly different. In HAR, is not unusual to have significantly different sizes of labeled samples for different subjects.

5.2.1 Formalisation Of The problem in FL settings for HAR

Given a non-IID set of datasets D, a standard *centralized* ML approach builds a recognition model M^C by using all the annotated data points in $D^* = D_1 \bigcup D_2 ... \bigcup D_n$. In this case, the training phase consists in finding the parameters $\mathbf{w} \in \mathbb{R}^{\mathbf{d}}$ that minimize a global objective function $f(\mathbf{w})$:

$$\min_{\mathbf{w}\in\mathbb{R}^{\mathbf{d}}} f(\mathbf{w}),\ f(\mathbf{w}) := \frac{1}{|D^*|}\sum_{k=1}^{|D^*|}\ell_k(\mathbf{w}) \tag{5.1}$$

where $\ell_k(\mathbf{w})$ is a loss function. Intuitively, the objective is to find the parameters $\mathbf{w}$ that minimize the average loss over all the annotated samples in D^*. By considering all the annotated samples at the same time, this *centralized* approach mitigates the non-IID problem.

However, there are significant differences in an FL setting. Indeed, each user U_i locally trains a model M_i, and it transmits to the server only the M_i parameters $\mathbf{w_i}$. The server is in charge of building a global model $\overline{M}$ from the local parameters $\mathbf{W} = \langle \mathbf{w_1}, \ldots, \mathbf{w_n} \rangle$, and it is not possible to directly access D^*. The objective function $\overline{f}(\overline{w})$ of the federated model to derive the global parameters $\overline{w}$ is the following:

$$\min_{\overline{\mathbf{w}}\in\mathbb{R}^{d}} \overline{f}(\overline{\mathbf{w}}),\ \overline{f}(\overline{\mathbf{w}}) = \sum_{i=1}^{n}\frac{|D_i|}{|D^*|}f_i(\mathbf{w_i}) \tag{5.2}$$

where $f_i(\mathbf{w_i})$ is the local objective function that each user U_i minimizes by using D_i to obtain $\mathbf{w_i}$:

$$\min_{\mathbf{w_i}\in\mathbb{R}^{d}} f_i(\mathbf{w_i}),\ f_i(\mathbf{w_i}) := \frac{1}{|D_i|}\sum_{k=1}^{|D_i|}\ell_k(\mathbf{w_i}) \tag{5.3}$$

Ideally, the parameters of the *federated* model should approximate the ones of the *centralized* model. However, in a non-IID setting, the overall data distribution of D^* (that is captured by the *centralized* approach) may be considerably different from the distribution of each $D_i \in D$ that is captured by the *federated* approach. For this reason, minimizing $\overline{f}(\mathbf{w})$ may lead to a global model that would significantly underperform the one derived by minimizing $f(\mathbf{w})$.

5.2.2 The Federated Clustering issue considering non-IID data

A possible solution to tackle the non-IID problem in the FL setting issue is to partition U into s clusters $C = C_1, ..., C_s$ so that each cluster minimizes the

non-IID properties among the datasets of the users assigned to the same cluster. Hence, it is possible to derive a federated model $\overline{M}^{C_j}$ for each cluster. The objective function $\overline{f}^{C_j}(\overline{\mathbf{w}}^{C_j})$ of each model M^{C_j} can be optimized by using data from the cluster:

$$\min_{\mathbf{w}^{C_j} \in \mathbb{R}^d} \overline{f}^{C_j}(\overline{\mathbf{w}}^{C_j}),\ \overline{f}^{C_j}(\overline{\mathbf{w}}^{C_j}) = \sum_{i=1}^{|C^j|} \frac{|D_i|}{|D^{C_j}|} f_i(\mathbf{w_i}) \tag{5.4}$$

where D^{C_j} is the set of datasets of the users belonging to the cluster C_j. If the clusters actually capture the similarity between the distributions of the datasets, the resulting model would better approximate the one generated by a *centralized* approach on the users of the same cluster.

However, in the FL setting it is not possible to access each D_i to compute the clusters, since only the model parameters $\mathbf{w_i}$ are available. Hence, a major problem that we tackle in this work is how to compute user clustering in the FL setting.

5.2.3 Data scarcity assumptions

In this chapter, we assume that the users do not actually have labeled datasets, but that they can only observe a stream of unlabeled sensor data. Let $S_i = \{x_1, x_2, \dots\}$ be the stream of unlabeled data points observed by U_i. We also assume that, given a data point x, a user U can sometimes provide feedback about the corresponding activity. Finally, we assume the existence of a small pre-training labeled dataset D^{pt} that can be used to initialize the federated model. Note that the data points in D^{pt} are not collected from the users in U.

The problem tackled in this chapter is to compute user clustering under the above-mentioned assumptions.

5.3 SS-FedCLAR: Combining Federated Clustering and Semi-Supervised Learning

5.3.1 Overview

We assume that each user that participates in SS-FedCLAR has a personal trusted device (e.g., a smartphone, a smartwatch, a smart-home gateway) that we will refer as *client*. The client is in charge of collecting sensor data and running the client-side SS-FedCLAR's algorithms.

At the very beginning, SS-FedCLAR uses a classic FL solution that is based on a single global model. Since we assume limited availability of labeled data, this model is initialized using a small labeled dataset called *pre-training dataset*. Considering real-world deployments, the *pre-training dataset* may be one or more public datasets, or a small training set specifically collected by appropriately rewarded volunteers. Since we assume that the clients do not have personal labeled datasets to train their local models, SS-FedCLAR relies on a semi-supervised learning strategy that analyes the periodic classifier's outputs to provide pseudo-labels to the stream of unlabeled data. The newly labeled data are then used to train the local model during FL global model updates. At the end of each FL training process, each client also uses a fine-tuning strategy based on transfer learning to further personalize the global model. Periodically, during global model updates, SS-FedCLAR uses a server-side Federated Clustering algorithm (explained in detail in Section 5.3.2) to group the clients in clusters based on the similarity of their local model updates, and to incrementally compute a specialized model for each cluster.

Since some clients may not belong to clusters due to peculiarity in their activity execution, SS-FedCLAR also considers *non-clustered clients*. Federated Clustering is transparent to clients, that are not aware if they belong to a cluster.

After clusters are finalized, the server uses a standard aggregation method to update each specialized cluster model using the local model updates received by the clients of that cluster. The server also maintains a classic global model

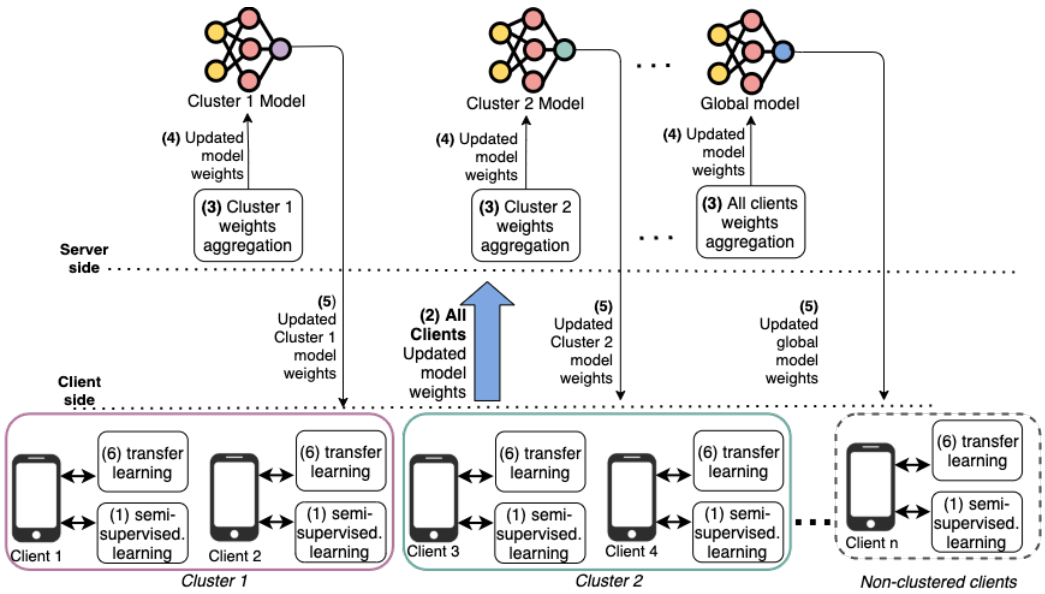


Figure 5.1: Overall architecture of SS-FedCLAR

using the updates of all clients, including *non-clustered clients* that will be the ones receiving this model during the updates. The overall architecture of SS-FedCLAR is summarized in Figure 5.1.

5.3.2 Server Side: Federated Clustering

We first describe the tasks performed by the server-side component of SS-FedCLAR.

Computing similarity between users

SS-FedCLAR adopts a server-side clustering approach to create specialized global models for groups of similar users. In general, clustering methods rely on a similarity metric that is computed on each pair of items that may be clustered. In sensor-based HAR, similar users are those that share similar sensor data patterns (i.e., similar activity patterns). However, in an FL learning process, only the weights of the local models are available, and not sensor data. Nonetheless, if two local models share similar weights, they were likely trained with similar patterns of data. Hence, given the parameter vectors $\mathbf{w_i}$ and $\mathbf{w_j}$ of the models corresponding to the users U_i and U_j, it is possible to compute their similarity.

SS-FedCLAR relies on the cosine similarity since it proved to be effective for federated clustering [111]. The cosine similarity between the model weights of two users U_i and U_j can be computed as follows:

$$sim(\mathbf{w_i}, \mathbf{w_j}) = \frac{\mathbf{w_i} \cdot \mathbf{w_j}}{\|\mathbf{w_i}\|\|\mathbf{w_j}\|} \tag{5.5}$$

However, considering HAR models and the recent results on transfer learning [23], we realized that computing the similarity by taking into account the whole parameter vector would not be the optimal choice. Considering local models based on deep learning, the closest layers to the input reflect high-level features that are common among all the subjects [107]. On the contrary, the layers that are close to the output are the ones that encode user-specific activity patterns.

Let $pers(\mathbf{w})$ be a function that extracts from parameter vector $\mathbf{w}$ the user-specific parameters. Hence, SS-FedCLAR computes the pairwise similarity between model weights as follows:

$$sim(\mathbf{w_i}, \mathbf{w_j}) = \frac{pers(\mathbf{w_i}) \cdot pers(\mathbf{w_j})}{\|pers(\mathbf{w_i})\|\|pers(\mathbf{w_j})\|} \tag{5.6}$$

Since SS-FedCLAR is based on deep learning, the function $pers(\mathbf{w})$ returns the weights corresponding to the last l layers of $\mathbf{w}$.

Hierarchical Clustering

Using the similarity function described above, the cloud server in SS-FedCLAR can apply a clustering algorithm to derive groups of users that perform activities in a similar way. In this work, we use a hierarchical approach, since in the literature it proved to be effective for federated clustering [109].

The pseudo-code for the hierarchical clustering method of SS-FedCLAR is described in Algorithm 4. The intuition behind this process is the following. Initially, there is a cluster for each user. Then, clusters are grouped based on the pairwise similarity of the participating users and a clustering threshold ct. When two clusters are merged into a single one, a new specialized model for that cluster is generated by merging the models of the merged clusters. The process is repeated until no more clusters can be merged (i.e., there is no pair of clusters

such that the similarity of their specialized models is higher than ct). The users in the singleton clusters are considered as *non-clustered clients*.

Algorithm 4 HierarchicalClustering

Input: $\mathbf{W} = \{\mathbf{w_1}, \ldots, \mathbf{w_n}\}$
Output: A set of clusters C, a set of specialized models $\mathbf{W^C}$

$\mathbf{W^C} \leftarrow \mathbf{W}$
$C \leftarrow \{\{U_1\}, \ldots, \{U_n\}\}$
$cmap \leftarrow$ empty map from model weights to clusters
$cmap[\mathbf{w_1}] \leftarrow \{U_1\}$
...
$cmap[\mathbf{w_n}] \leftarrow \{U_n\}$
do
    $P \leftarrow$ pairwise similarity matrix on $\mathbf{W^C}$ based on $sim()$
    $\mathbf{w_a}, \mathbf{w_b} \leftarrow \underset{\mathbf{w_a}, \mathbf{w_b} | a \neq b}{\arg\min}\ P_{ab}$
    if $sim(\mathbf{w_a}, \mathbf{w_b}) \geq ct$ **then**
        $\mathbf{w_{ab}} \leftarrow$ merge $\mathbf{w_a}$ and $\mathbf{w_b}$ using FedAvg
        $C_a \leftarrow cmap[\mathbf{w_a}]$
        $C_b \leftarrow cmap[\mathbf{w_b}]$
        $C_{ab} \leftarrow C_a \cup C_b$
        $cmap[\mathbf{w_{ab}}] \leftarrow C_{ab}$
        $C \leftarrow C \setminus C_a$
        $C \leftarrow C \setminus C_b$
        $C \leftarrow C \cup C_{ab}$
        $\mathbf{W^C} \leftarrow \mathbf{W^C} \setminus \mathbf{w_a}$
        $\mathbf{W^C} \leftarrow \mathbf{W^C} \setminus \mathbf{w_b}$
        $\mathbf{W^C} \leftarrow \mathbf{W^C} \cup \mathbf{w_{ab}}$
    else
        $\mathbf{W^C} \leftarrow \{\mathbf{w} \in \mathbf{W^C}\ such\ that\ |cmap(\mathbf{w})| > 1\}$
        $C \leftarrow \{C_j \in C\ such\ that\ |C_j| > 1\}$
        return C and $\mathbf{W^C}$
    end if
while True

Model Update in SS-FedCLAR

The model update mechanism of SS-FedCLAR is described by Algorithm 5. Periodically (e.g., every night), the server requires an update of the models. Hence, a sequence of communication rounds is started. Each client locally trains its model and transmits the resulting weights to the server. Upon receiving the weights from the clients, the first task of the server is generating an overall global model using FedAvg.

If required, during the update the server computes clusters and specialized models as described before. Note that computing the similarity between users is effective only if performed after a certain number of communication rounds. Otherwise, we experimentally observed the risk of considering model parameters that are not sufficiently trained, thus generating unreliable clusters. For this reason, in SS-FedCLAR, our hierarchical clustering method explained in Section 5.3.2 is performed after a predefined number r of communication rounds. From the communication round r, the server will use the local model updates received from the clients to update the specialized models, while the ones received from the *non-clustered clients* are used to update the overall global model. Note that, in order to provide to *non-clustered clients* a global model with sufficient generalization capabilities, also the local models from clustered clients are used to update the overall global model.

Based on preliminary experiments, we observed that it is not necessary to perform clustering at each model update if the set of participating users and their local data distribution do not change. We hypothesize that it may be necessary to introduce the clustering step in the model update only when there is a significant change. A deeper investigation of this aspect is out of the scope of this paper, and we discuss possible solutions in Section 7.2.

5.3.3 Client Side: Semi-Supervised Learning

In SS-FedCLAR, the clients do not have a labeled dataset to train the local model. In order to overcome this problem, we proposed a semi-supervised strategy inspired by the one presented in Section 4.2.8 of Chapter 4. In particular, each client semi-automatically provides pseudo-labels to the unlabeled data stream by combining *Active Learning* and *Label Propagation*.

Classification and active learning

As we previously mentioned, active learning involves asking the user the activity label only for those data samples where the classifier shows uncertainty in activity classification. Indeed, the data samples associated with low classification confidence would have the most impact on improving the activity model if their

Algorithm 5 SS-FedCLAR - Server Side Model Update

```
C ← nil
W^C ← nil
for each periodic update (e.g., every night) do
    for each communication round i do
        receive W={w_1, .., w_n} from clients
        w^G ← FedAVG using W (global model)
        if i < r and W^C == nil then
            send w^G to each client
        else
            if i == r and cluster update is required then
                C, W^C ← HierarchicalClustering(W)
            else
                for C_j ∈ C do /*for each cluster*/
                    w_{C_j} ← FedAVG using w_i ∈ W from clients in C_j
                end for
            end if
            for C_j ∈ C do
                send w_{C_j} to each client in C_j
            end for
            send w^G to non-clustered clients
        end if
    end for
end for
```

label were available. The overall pseudo-code of classification and active learning strategy of SS-FedCLAR is detailed in Algorithm 6.

Label Propagation and Model Update

During a model update, the clients' first step is to start a Label Propagation process in charge of expanding the amount of labeled data. Besides training the local model with more labeled examples, Label Propagation has the major advantage of further reducing active learning queries, and hence interactions with the user. Given a set of labeled and unlabeled data points, the Label Propagation algorithm automatically spreads pseudo-labels to a portion of unlabeled data [143]. Intuitively, data points that are close in the feature space likely correspond to the same class label. SS-FedCLAR adopts the Label Propagation strategy presented in Section 4.2.8.

After Label Propagation, the actual model update is started: for each com-

Algorithm 6 Client side - Classification and active learning

```
1: lm ⇐ local model
2: for each feature vector fv computed in real-time from sensor data do
3:     p⃗ ⇐ probability distribution over the activities predicted by lm on fv
4:     output the most likely activity according to p⃗
5:     if feedback is needed according to VAR-UNCERTAINTY [132] then
6:         l ⇐ activity label from the user
7:         add (fv, l) to the Feature Vectors Storage
8:         update threshold of VAR-UNCERTAINTY [132]
9:     else
10:        add (fv, −) to the Feature Vectors Storage          ▷ unlabeled data point
11:    end if
12: end for
```

munication round, the clients train their local model with the available labeled data samples in the *Feature Vectors Storage*. Then, the resulting weights are transmitted to the server[1]. Finally, the clients receive the updated global model from the server. Each client is not aware if it is receiving updated weights from specialized cluster models or from a generic global model.

Even though Federated Clustering mitigates the non-IID problem, it is still possible that distinct users in the same cluster exhibit peculiar execution of some activities. Indeed, while some users may be grouped in the same cluster because they similarly perform the majority of the activities, they still may exhibit slight differences over a restricted number of activities.

In order to further personalize the recognition model, SS-FedCLAR also relies on a fine-tuning approach for each client, inspired by transfer learning solutions that proved to be effective in FL-based HAR [23]. In particular, the last p layers (i.e., the closest to the output) of the local model are fine-tuned using the *Feature Vectors Storage*, while the remaining ones are left as received by the server.

The client-side model update process of SS-FedCLAR is summarized is by Algorithm 7.

[1]Each client also transmits the number of labeled data points used to train the local model. This information is needed for the FedAVG algorithm.

Algorithm 7 Client side - Model Update

1: $lm \leftarrow$ local model
2: Update the *Feature Vectors Storage* using the Label Propagation algorithm [147]
3: **for** each communication round **do**
4: train lm using available labeled data in the *Feature Vectors Storage*
5: send the weights $\mathbf{w}$ of lm to the server
6: receive updated model $\mathbf{w^S}$ from the server
7: replace the weights of lm with $\mathbf{w^S}$
8: **end for**
9: freeze the layers of lm except for the last p layers
10: train lm using available labeled data
11: unfreeze lm layers

5.4 Experimental Evaluation

In order to evaluate the effectiveness of FedCLAR, we considered two well-known HAR datasets: WISDM [51] and MobiAct [121]. More details about these datasets are given in Section 2.3. It is important to note that we used WISDM and MobiAct since they include a relatively large number of subjects with respect to other sensor-based HAR datasets. Even though a real deployment would involve a significantly larger number of participants, this aspect is crucial to evaluate our FL-based approach, considering that data (and participant) augmentation techniques may not lead to realistic results. Moreover, the subjects that participated in data collection in these datasets exhibit both data and label distribution skew, which is necessary to evaluate the clustering capabilities of SS-FedCLAR.

WISDM includes labeled activity data from 36 different subjects obtained from the accelerometer of a smartphone placed in the pants pocket during the activity execution. The activities considered in this dataset are: *walking*, *jogging*, *sitting*, *standing*, and *taking stairs*. The MobiAct dataset includes labeled activity data from 60 different subjects. Those data were collected from the inertial sensors (i.e., accelerometer, gyroscope, and magnetometer) of a smartphone placed in the pants pocket. In our experiments, we considered the following physical activities *standing*, *walking*, *jogging*, *jumping*, *going upstairs*, *going downstairs*, and *sitting*.

5.4.1 Experimental setup

As activity model, in our experiments, we used a simple feed-forward deep neural network composed of three fully connected layers having respectively 32, 16, and 16 neurons, and a softmax layer for classification. The inputs of the network are hand-crafted feature vectors extracted in real-time from the stream of sensor data. We consider features that proved to be effective for HAR in the literature [37]. We used Adam as optimizer. Even though existing FL approaches proposed more sophisticated deep learning classifiers (even to collaboratively learn feature representation), a simpler model with hand-crafted features allowed us to focus only on the specific semi-supervised clustering problem. Moreover, we believe that an advantage of our simple model is a reduced computational effort, which is more suitable for mobile devices. Some of the hyper-parameters were selected considering the results of our previous work [148]: $l = 1$, $p = 2$, $r = 5$, and 10 local training epochs with a batch size of 30 samples. The remaining hyper-parameters were selected using a grid search, with the objective of optimizing the overall F1 score. For instance, considering the clustering threshold ct, we chose $ct = 0.0035$ for the WISDM dataset, and $ct = 0.0030$ for the MobiAct dataset. The impact of the clustering threshold on the recognition rate and quality of clusters is reported in Section 5.4.3.

5.4.2 Evaluation methodology

Since we consider a semi-supervised approach, we decided to use an evaluation methodology that shows the evolution of the recognition rate and the number of active learning queries. In particular, we adapted the evaluation methodology proposed in Section 4.3.1 to include Federated Clustering.

First, we split the dataset into two partitions called Pt and Tr. The partition Pt (i.e., pre-training data) includes data from users that are only used to initialize the global model. The partition Tr (i.e., training data) includes data of the users who actually participate in the FL process. In our experiments, we randomly partition the users into 15% whose data will populate Pt, and 75% whose data will populate Tr. Moreover, we partition the data for each user in

Tr into sh shards having the same size. Intuitively, a shard represents the time period that separates two model updates. Unfortunately, the considered datasets have a limited amount of data not temporally distributed in shards (e.g., day). Moreover, shards will more likely have similar data distributions among the different activities. Hence, we randomly assign to each shard of a user u a fraction $\frac{1}{sh}$ of the available u's data samples, making sure of reflecting in each shard the distribution of the data samples. This approach allows us to mimic a realistic scenario where users perform several types of activities in each shard.

In the following, we describe our evaluation strategy in detail. The global model (pre-trained with Pt) is first distributed to the clients of the users in Tr, which will use it as the first version of the *local model*. Then, the process is composed of sh iterations, one for each shard. During a shard, each device exploits the current *local model* to classify the continuous stream of sensor data. The classification output is used to evaluate the recognition rate in terms of the F1 score. In parallel, we also apply our active learning strategy keeping track of the number of triggered questions. At the end of the shard, our Federated Learning process starts and the local models are updated. Since for each user the data distribution in its shards was similar, the hierarchical clustering algorithm is performed only at the first shard. In Section 5.4.3 we show the impact of clustering at different shards.

In our experiments, we empirically determined $sh = 4$ for the WISDM dataset and $sh = 3$ for the MobiAct Dataset. Higher values of sh would lead to an insufficient amount of data samples in each shard, thus negatively impacting the evaluation of SS-FedCLAR.

5.4.3 Results

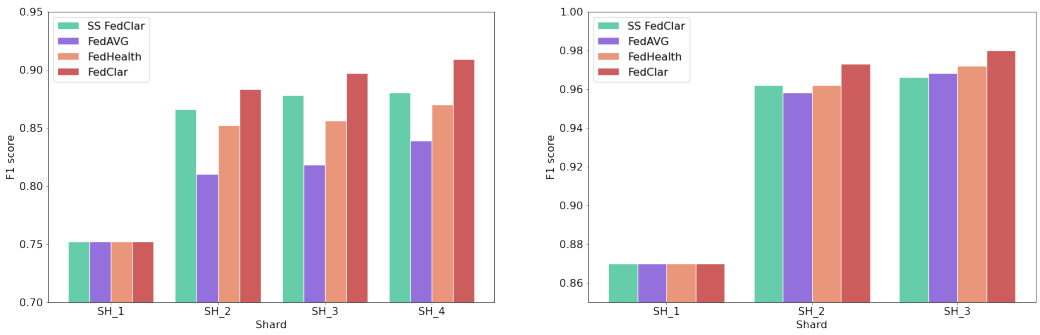
In the following, we report the results of our evaluation. In particular, we compared SS-FedCLAR considering four baselines:

- **FedAvg** [19]. A classic fully supervised FL approach not considering the non-IID problem.

- **FedHealth** [23]. A fully-supervised FL-based approach for HAR that tackles the non-IID problem using transfer learning.
- **FedCLAR** [148]. A fully-supervised Federated Clustering method for HAR.
- **FedAR** [147]. The semi-supervised FL approach for HAR that we introduced in Chapter 4. It combines semi-supervised learning with a fine-tuning strategy inspired by transfer learning to mitigate both the labeled data scarcity issue and the non-IID problem.

Overall recognition rate

Figure 5.2 shows, on both datasets, the recognition rate of SS-FedCLAR at each shard. This figure also compares SS-FedCLAR with the fully-supervised baselines (assuming that clients have complete availability of labeled data at each shard). Even though SS-FedCLAR uses a limited amount of labeled data, with respect to FedCLAR it is only $\approx 1.5\%$ behind on the MobiAct dataset and $\approx 2.6\%$ on WISDM. Considering the remaining baselines, SS-FedCLAR outperforms them on the WISDM dataset, while it reaches similar results on the MobiAct dataset.



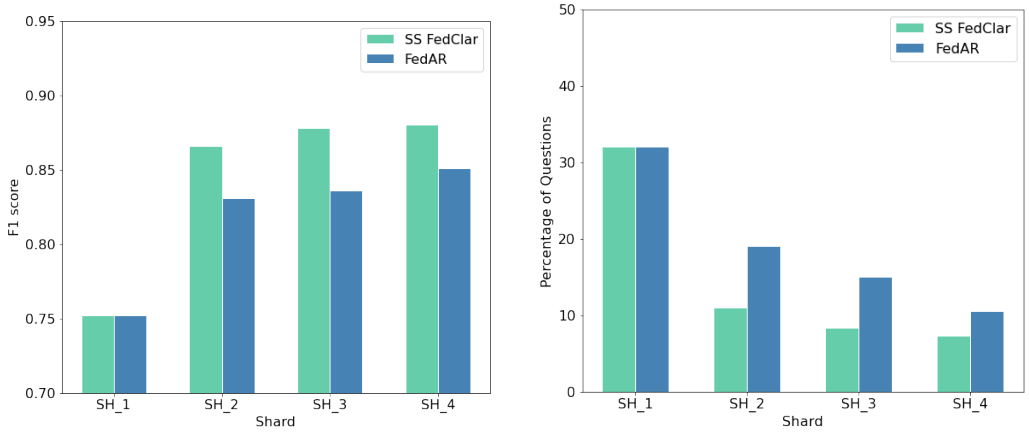
(a) WISDM (b) MobiAct

Figure 5.2: SS-FedCLAR vs. fully supervised baselines shard by shard (F1 score)

These results show that our method is capable of reaching results that are close to the state-of-the-art approaches without assuming labeled data availability.

Figure 5.3 and Figure 5.4 compare SS-FedCLAR with FedAR, another FL-based semi-supervised HAR approach based on semi-supervised and transfer

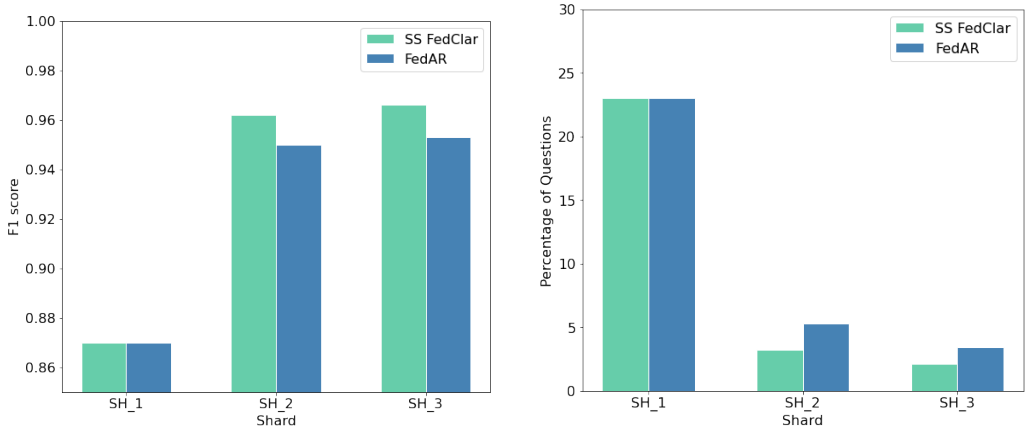
learning. The results indicate that SS-FedCLAR reaches higher recognition rates



(a) Average F1 score (b) Average percentage of questions

Figure 5.3: **WISDM**: Comparison shard by shard of SS-FedCLAR with FedAR in terms of the F1 score and the percentage of triggered questions

on each shard with respect to FedAR and, at the same time, it triggers a significantly lower number of active learning questions. This is due to the fact that our Federated Clustering algorithm better mitigates the non-IID problem, reducing uncertainty during classification.



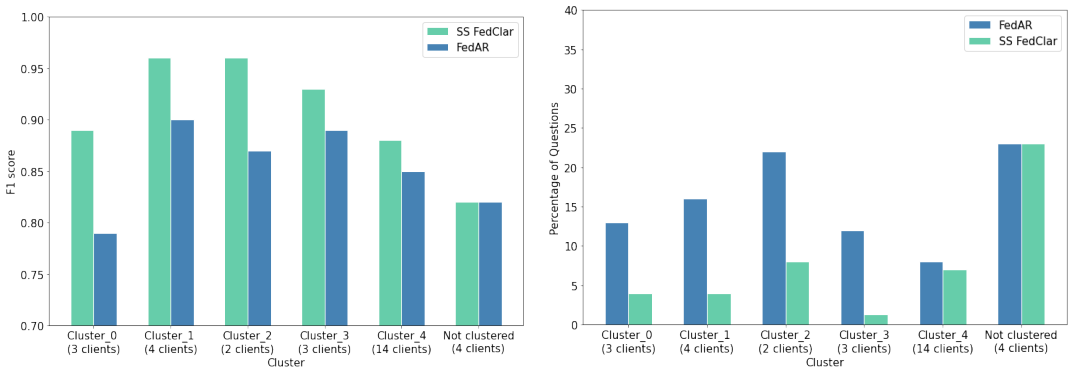
(a) Average F1 score (b) Average percentage of questions

Figure 5.4: **Mobiact**: Comparison shard by shard of SS-FedCLAR with FedAR in terms of the F1 score and the percentage of triggered questions

Cluster-based results

Figure 5.5 and Figure 5.6 show the results of SS-FedCLAR at the cluster level. For each cluster generated by SS-FedCLAR, we compare the F1 score and the percentage of active learning questions of SS-FedCLAR with the ones of FedAR. Note that these results are just a detailed version of the ones proposed in Figure 5.3 and Figure 5.4. Indeed, FedAR is actually evaluated considering all the users, while we show the F1 score considering the subsets of the users based on the output of SS-FedCLAR.

Our results show that the federated clustering method has a positive impact on each cluster, especially considering the WISDM dataset. We also observed that only a small percentage of clients was not clustered. Since those clients use a general FL global model, the recognition rate and the number of active learning questions of SS-FedCLAR are similar to the ones of FedAR.



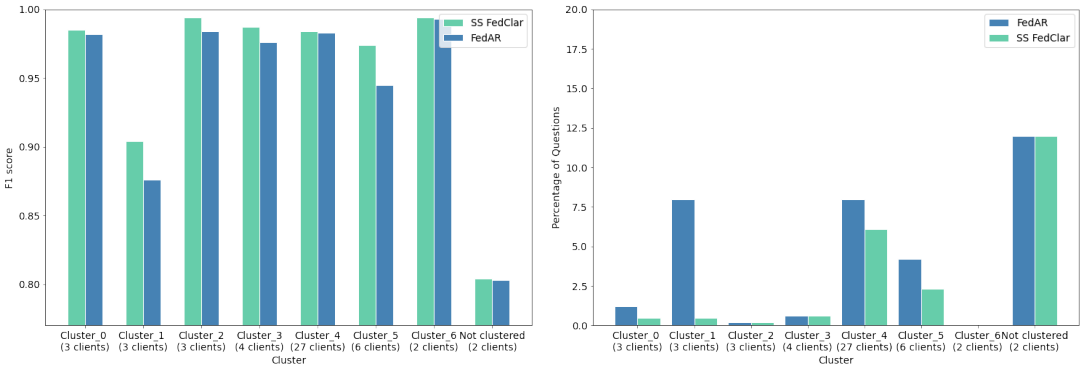
(a) Average F1 score

(b) Average percentage of questions

Figure 5.5: **WISDM**: Comparison of SS-FedCLAR with FedAR cluster by cluster in terms of F1 score and percentage of triggered questions

Impact of the clustering threshold

In the following, we show the impact of the clustering threshold *ct* on the recognition rate. Table 5.1 and Table 5.2 show that the choice of *ct* has a significant impact on the recognition rate, the number of clusters, and the percentage of not-clustered clients. When *ct* is too low, SS-FedCLAR generates small clusters



(a) Average F1 score

(b) Average percentage of questions

Figure 5.6: **Mobiact**: Comparison of SS-FedCLAR with FedAR cluster by cluster in terms of F1 score and percentage of triggered questions

ct	F1	# clusters	clients not clustered
0.0010	0.83	6	56.67%
0.0020	0.85	8	20.00%
0.0030	0.85	6	16.67%
0.0035	**0.88**	**5**	**13.33%**
0.0040	0.86	4	10.00%
0.0045	0.85	4	6.67%
0.0050	0.85	4	6.67%

Table 5.1: WISDM: Impact of the clustering thresholds

ct	F1	# clusters	clients not clustered
0.0010	0.94	11	23.53%
0.0020	0.95	8	11.76%
0.0030	**0.96**	**7**	**3.92%**
0.0035	**0.96**	**7**	**3.92%**
0.0040	**0.96**	**7**	**3.92%**
0.0045	0.95	6	3.92%
0.0050	0.94	5	3.92%

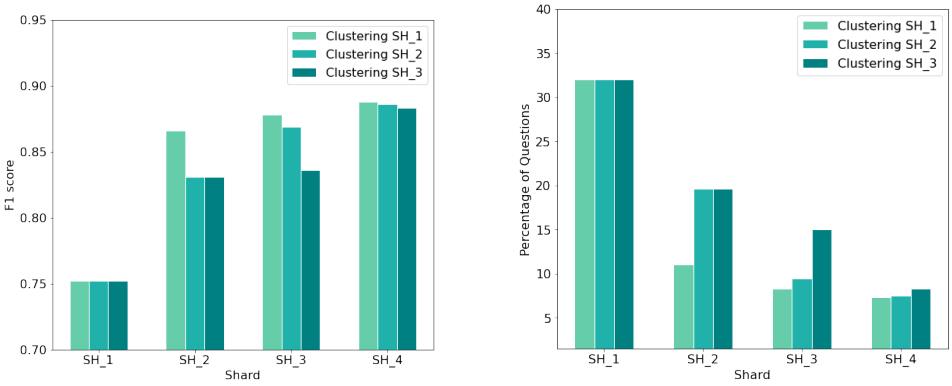
Table 5.2: MobiAct: Impact of the clustering thresholds

and a high rate of *not-clustered* clients, thus negatively impacting the recognition rate.

By closely inspecting the results on the MobiAct dataset in Table 5.2, we noticed that *ct* values higher than 0.003 do not change the percentage of *not-clustered clients*. This is likely due to the fact that, in this dataset, the users corresponding to not-clustered clients perform activities in a very different way with respect to all the other users.

The impact of clustering at different shards

As we previously mentioned, due to the nature of the considered datasets, in our experiments we performed clustering only during the model update at the first shard. Figure 5.7 and Figure 5.8 show the impact of performing clustering at different shards on both datasets. We observed that the advantage of performing clustering at the first shard is that it immediately improves the recognition rate in the following shards and, at the same time, it quickly reduces the number of active learning questions. Hence, these results indicate that even if the clusters are created at the very first shard, they are reliable. This is likely due to the fact that data distribution in the different shards does not change significantly.



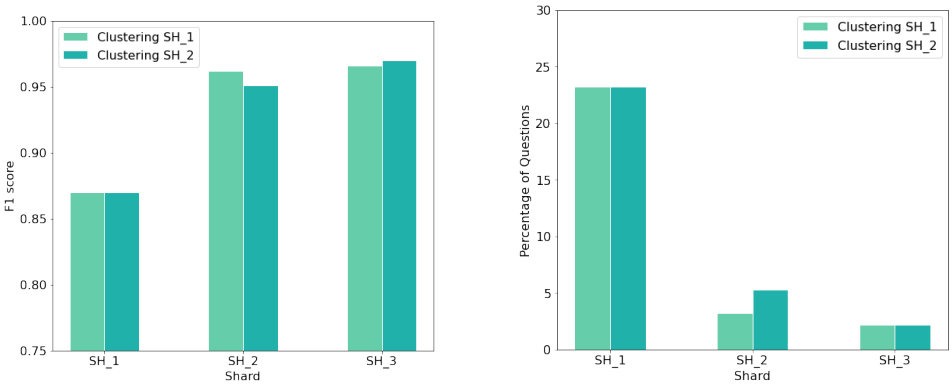
(a) Average F1 score

(b) Average percentage of questions

Figure 5.7: **WISDM**: The impact of clustering at different shards.

The impact on non-IID data

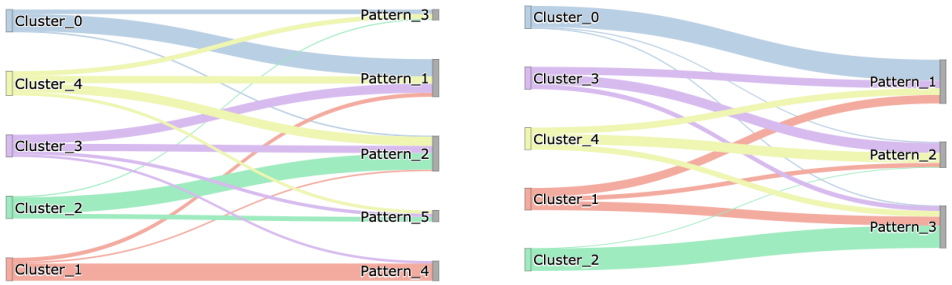
In the following, we show how the non-IID problem is actually mitigated by SS-FedCLAR on the considered datasets. First, we investigate the feature distribution skew. This condition occurs when different users perform the same activity with different patterns. We expect that users grouped in the same cluster perform activities in a similar way, while users in different clusters execute activities in different ways. In order to evaluate if the clusters generated by SS-FedCLAR have this property, from the raw sensor data of all users in each dataset we extract, for each activity, a set of patterns. Each pattern characterizes a way



(a) Average F1 score (b) Average percentage of questions

Figure 5.8: **Mobiact**: The impact of clustering at different shards.

of performing that activity [2]. Then, we correlate the patterns with the clusters of users generated by SS-FedCLAR. For the sake of brevity, we report a couple of examples related to the WISDM dataset in Figure 5.9.



(a) Going Downstairs (b) Jogging

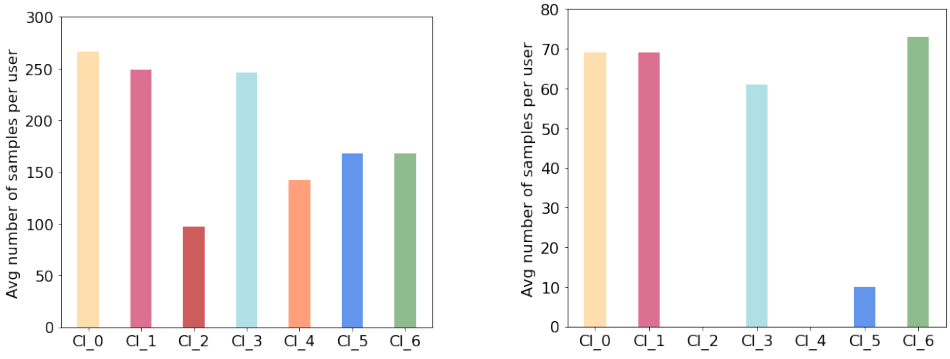
Figure 5.9: WISDM: examples of feature distribution skew. The plot shows the correlation between clusters generated by SS-FedCLAR and activity patterns.

From this analysis, it emerges that many clusters generated by SS-FedCLAR in WISDM exhibit a peculiar correlation with activity patterns. Hence, the non-IID problem is reduced with a positive impact on the recognition rate. For instance, considering the activities in Figure 5.9, the improvement in overall

[2]We normalize raw sensor data, we apply PCA for dimensionality reduction and we apply the K-Means algorithm. In order to find the optimal number of clusters for each activity, we maximize the Silhouette score.

F1-score of SS-FedCLAR with respect to FedAR is +12% for *going downstairs* (from 0.64 to 0.76), while +4% for *jogging* (from 0.90 to 0.94). We observed an improvement in the F1 score for each activity in WISDM whenever there is a clear correlation between clusters and patterns.

We also noticed that the feature distribution skew does not clearly emerge in MobiAct, since most of the users in this dataset tend to perform activities with similar patterns. This is consistent with the results presented above: SS-FedCLAR has in general a minor improvement on this dataset with respect to WISDM. The improvement of SS-FedCLAR on MobiAct is still appreciable since, differently from WISDM, this dataset suffers from a label distribution skew. Hence, SS-FedCLAR is still able to improve the recognition rate by grouping users that have similar label distributions. Figure 5.10 shows this property for a couple of activities. Considering the examples in this figure, the improvement in F1-score of SS-FedCLAR with respect to FedAR is +5% (from 0.92 to 0.97) for *walking*, while +7% for *sitting* (from 0.86 to 0.93). We observed an improvement in the F1 score for each activity in MobiAct whenever there is a clear correlation between clusters and skewed label distributions.



(a) Walking (b) Sitting

Figure 5.10: MobiAct: examples of labels distribution skew. The plot shows the average number of activity samples for each user in the clusters generated by SS-FedCLAR.

5.5 Summary

In this Chapter, we presented SS-FedCLAR, a novel solution that combines federated clustering and semi-supervised learning for HAR. Our results show that SS-FedCLAR addresses the research question **Q3)** presented in Section 2.4, by strongly reducing the non-IID problem typical of Federated HAR, while tackling the labeled data scarcity issue as well. Indeed, SS-FedCLAR reaches recognition rates that are very close to fully-supervised methods and it outperforms state-of-the-art semi-supervised FL-based HAR approaches.

Among the limitations of the current version of SS-FedCLAR, there is the potential leak of private information to an honest-but-curious service provider running the server infrastructure. It is well known that, despite only model parameters being shared with the server, some personally identifiable data could be still inferred from them. In order to mitigate this issue, FL approaches usually rely on Secure MultiParty Computation (SMC) to aggregate the local weights in a privacy-preserving fashion [149]. SMC makes it possible to hide from the service provider the mapping between each local model and the corresponding subject. Even when this type of protection is applied, FL models are exposed to several types of attacks that extract private information from the global model parameters [150]. Examples of such attacks are the reconstruction attack [151], the membership inference attack [152], and the property inference attack [153].

In order to explore these privacy vulnerabilities related to FL for HAR, in the following Chapter we will introduce a novel methodology that uses the membership inference attack to assess the potential privacy leakages of a global activity recognition model.

Chapter 6

Sensitive Data Leakage in Federated Human Activity Recognition

6.1 Introduction

In the previous Chapters, we introduced FedAR and SS-FedCLAR, two novel collaborative approaches for sensor-based human activity recognition that take advantage of the Federated Learning (FL) framework to mitigate privacy issues related to collaborative HAR [19, 154]. However, despite FL avoiding the release of labeled sensor data, recent studies show that the model's parameters received and manipulated by the cloud server may still reveal sensitive information about the data used by FL users to train the recognition model [26]. While several research works recently investigated how to infer sensitive information from FL models in various domains, to the best of our knowledge the potential privacy leakages of federated HAR models have not been studied yet.

In this Chapter, we make the first step along this line of research by proposing a novel framework to quantitatively measure the potential information leakage of the global models' weights in federated HAR. Our framework relies on the Membership Inference Attack (MIA) [27] to try inferring the following sensitive

information about participating users: a) whether a specific subject is one of the FL participating users, b) whether a specific participant contributed to the global model with a particular activity. It is also important to note that here we decided to consider a traditional FL-based setting for HAR instead of the semi-supervised or clustering-based approaches that we presented in the previous sections. We made this choice to provide a general overview of the potential privacy leakage in Federated HAR without introducing bias related to our specific solutions, such as the use of a limited number of training samples to learn the model or the presence of personalized cluster models server-side.

The obtained preliminary experimental evaluation suggests that it is possible to derive sensitive information from HAR global models. Hence, we hope that these results may pave the way to further research investigations in this area.

6.2 Membership Inference Attack in FL-based HAR

6.2.1 Membership Inference Attack

The objective of the Membership Inference Attack (MIA) is to infer whether a specific data sample has been used or not to train a DL model. Formally, given a set X of data samples (represented by feature vectors), let D^t be a labeled dataset of pairs (x, y) where $x \in X$ and y is a label. D^t is used to train a target model M^t.

MIA assumes the access to M^t and uses a binary classifier (i.e., the attack model) to determine if a data sample $x \in X$ appears in a pair (x, y) of D^t or not. In the first case, we say that x is a *member* data sample, while in the second case is a *non-member*. The attack model performs such classification by analyzing the behaviour [1] of M^t in classifying the feature vector x. Details about the construction of the attack model will be given in Section 6.3.1 considering the specific domain of HAR.

[1] Examples of relevant behaviours are the gradients' variations and the confidence of the model while classifying an input data.

6.2.2 Membership Inference Attack in FL

In FL, an attacker may perform the MIA attack on the global model (the target model M^t). Since the FL cloud service provider has no access to the training dataset D^t, the authors in [28, 112] proposed to train the attack model using a *shadow model* trained with a *shadow training dataset*.
A *shadow model* M^s aims at imitating the behaviour of M^t. In particular, the attacker creates a pair of disjoint *shadow training sets* D^s (members shadow data) and N^s (non-members shadow data), such that each training set contains labeled data samples in the same feature and label space as D^t. Moreover, these training datasets should have a similar distribution to D^t. In practice, shadow training datasets can be obtained by public datasets or by generating synthetic data.

M^s is trained by using D^s, and the attack model is trained by analyzing the behaviour of M^s while classifying the data samples in D^s and N^s. The intuition is that, since both M^t and M^s are trained with data that share a similar data distribution, the attack model trained considering the behaviour of M^s in classifying members and non-members data samples would also be effective for M^t.

6.2.3 Shadow models for HAR

Considering the specific HAR domain, the generation of a shadow dataset D^s is particularly challenging. This is a well-known limitation of the attacks based on MIA: approximating the distribution of data strictly related to a specific set of individuals is challenging [153]. In HAR, due to the high intra- and inter-variability in activity execution among several subjects (i.e., each subject has peculiar activity patterns and habits), the underlying data distribution is not independent and identically distributed (non-IID). If D^s is significantly different from D^t, the attack performance of MIA degrades accordingly [155]. This problem becomes serious when D^t includes a large number of users with different characteristics. For this reason, we consider a worst-case scenario in which the attacker manages to use a shadow dataset D^s very close[2] to the actual training dataset D^t. Moreover, similarly to other applications of the MIA, we assume that the attacker has

[2]In the experiments this is implemented by taking $D^s \subset D^t$.

access to some data samples of the participating users to perform the attack.

6.3 The proposed attack framework

In the following, we propose a novel framework based on the Membership Inference Attack (MIA) to quantitatively measure the amount of sensitive information potentially revealed by the global model in FL-based HAR. In particular, we investigate two research questions:

- R_1) *User Membership*: Is it possible to infer from the global model whether a certain user took part in the FL process? This property may be crucial considering FL systems that are specialized for a certain category of users (e.g., subjects with the same disease).
- R_2) *Activity Membership*: Is it possible to infer from the global model whether a participating user performed a specific activity?

For the sake of this thesis, we only consider honest-but-curious attackers that infer sensitive data by periodically observing the parameters of the global model: the *cloud server* and the *participating users*.

6.3.1 Attack model training

Given the notation introduced in Section 6.2, in our setting $D^t = \{(x_1, y_1),$ $..., (x_n, y_n)\}$ is the set of labeled samples from all the participating clients, while M^t is the global model on the cloud server. M^t is trained with a FL approach. In order to perform the MIA attack, the attacker trains a binary classifier A to determine if a given data sample belongs to D^t. In particular, we take advantage of the attack model recently proposed in [27]. This attack assumes that the attacker can inspect the internal parameters of M^t. In our FL setting, this is actually possible. Figure 6.1 depicts a high-level data flow of the attack model training. In order to train A, the attacker creates the shadow datasets D^s and N^s, as well as a shadow model M^s trained using D^s. We recall that D^s has a similar distribution to D^t. Then, each data sample in D^s and in N^s is provided to M^s for classification. While processing each input, the attacker observes the

behavior of M^s. In particular, given an input x provided to the shadow model M^s, the attacker extracts:

- The confidence of M^s in classifying x
- The output of each layer of M^s while processing x
- The classification loss $\ell(M^s(x), y)$
- The gradients of the loss with respect to each parameter of M^t

These values are encoded in a feature vector, that is labeled as *member* if $x \in D^s$ and *non-member* if $x \in N^s$. The resulting labeled feature vectors are used to train A.

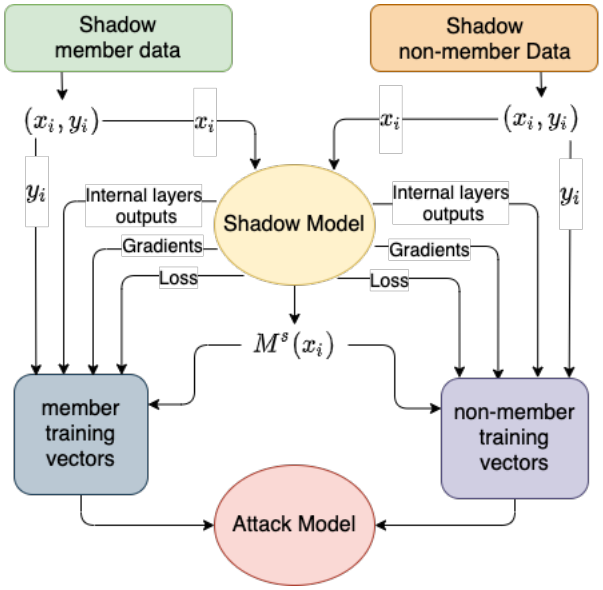


Figure 6.1: Training of the attack model. The attacker observes the behavior of the shadow model when classifying *member* and *non-member* data points. The output is the training dataset for the *attack model*.

6.3.2 Inferring user and activity membership

In the following, we illustrate how our framework infers user and activity membership using an attack model based on MIA.

Let $U = \{u_1, \ldots, u_n\}$ be a set of n users. In order to answer the research question R_1), we use the attack model to infer whether a certain user $u \in U$ contributed in training the global model. In this scenario, we assume that the attacker knows the corresponding user for each available data sample. The attacker infers that a subject u participated in training the global model if the majority of the samples of u tested by the attacker are classified as *members*. We quantitatively estimate the success of the attack by computing the average confidence of the attack model in classifying data samples of u as *members*.

In order to answer the research question R_2), we use the attack model to infer whether u participated in training the global model with an activity a. In this scenario, we also assume that the attacker knows the activity label for each available sample. The attacker infers that u participated in training the global model with activity a when the majority of the available labeled samples of u related to the activity a tested by the attacker are classified as *members*. We quantitatively estimate the success of the attack by computing the average confidence of these classifications.

6.4 Experimental Evaluation

6.4.1 Experimental setup

We perform a preliminary evaluation of our framework using the publicly available MobiAct dataset [156] introduced in Section 2.3.1. MobiAct includes labeled data from inertial sensors (i.e., accelerometer, gyroscope, and magnetometer) from a smartphone placed in the pant's pocket. Overall, MobiAct includes data from 60 subjects. In our experiments, we considered the following physical activities [3]: *standing*, *walking*, *jogging*, *jumping*, and *sitting*. Since this dataset involves a relatively large number of subjects with respect to other sensor-based HAR datasets, it is particularly suited to evaluate FL-based solutions. In our experiments, we consider a FL client for each user in MobiAct.

[3]Note that we omitted from MobiAct those physical activities with a limited number of samples as they are insufficiently represented and hence not suitable for our evaluation. We believe that this problem is only related to this specific dataset and that, in realistic settings, even short activities would be represented by a sufficient number of samples.

Federated Learning

We use the FL experimental setup proposed in Chapter 5 in Section 4.3.2, since it exhibited promising performances for HAR. In particular, the activity model is a feed-forward deep neural network composed of three fully connected layers having respectively 128, 64, and 32 neurons, and a softmax layer for classification. The inputs of that network are hand-crafted feature vectors extracted in real-time from the stream of sensor data. We consider features that proved to be effective for HAR in the literature [37]. We used Adam [145] as optimizer. The well-known FedAvg algorithm [19] is in charge of aggregating the model parameters received by clients and updating the global model. Each client trains its local model for 10 epoch. Finally, we empirically selected 30 as the number of FL communication rounds as it guarantees the convergence of the *global model* avoiding overfitting.

Membership Inference Attack

The implementation of MIA is based on the public *ML Privacy Meter*[4] tool [27]. For each experiment, we trained the *attack model* for 150 epochs with a learning rate of 0.001, while the Adam optimizer was used to minimize the loss function. As we mentioned in Section 6.2.3, in our experiments the shadow model is trained by using a subset of labeled data from the participating users.

Metric

In order to quantitatively measure the probability that a sample x was part of the target dataset D^t given a target model M^t, we use the confidence of the attack model in classifying x as *member*. We will refer to this measure as the *membership probability* (MP):

$$MP(x) = Pr(x \in D^t | M^t)$$

Intuitively, an MP value closer to 1 indicates that x is likely a *member*, while an MP value closer to 0 indicates that x is likely a *non-member*.

[4] https://github.com/privacytrustlab/ml_privacy_meter

6.4.2 Evaluating user membership

Data preparation

The data partitioning schema is illustrated in Figure 6.2. As usually proposed in FL methods, we randomly select 15% of the users from the dataset to initialize the global model (pre-training). The remaining users are partitioned as follows: 50% of users participate in FL (FL members) and 50% of users do not participate in FL (FL non-members) [5]. The global model is hence trained in a FL fashion using data in D^t. We train the attack model by using 70% of data from D^t labeled as *members*, and 70% from labeled as *non-members* [6]. We use the remaining 30% from both datasets to evaluate the effectiveness of the attack model.

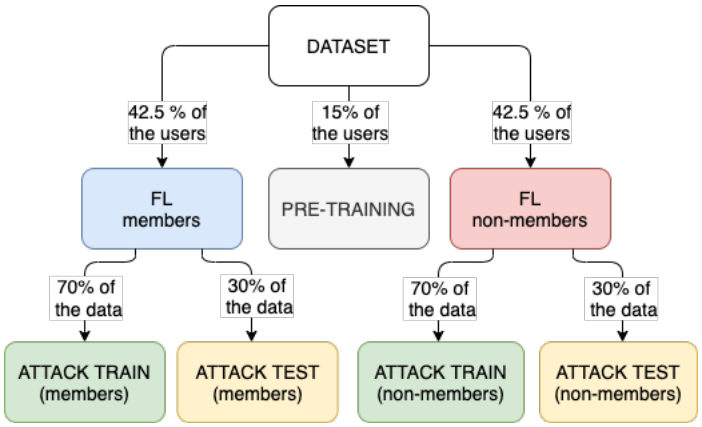


Figure 6.2: Dataset splitting process adopted to evaluate user membership

Results

Figure 6.3 shows the results of the user membership attack at the data sample granularity. We observed an MP value close or equal to 1 for most of the *FL members*' data samples, while a value close or equal to 0 for most of the *FL non-members*' samples. Thus, we can conclude that, overall, the *attack model* is confident in discriminating *members* and *non-members* samples.

Figure 6.4 shows the same result at the user granularity. In particular, for each user, we average the MP score computed on its test data sample. We can

[5]Note that the union of labeled data from *FL members* corresponds to D^t.
[6]Note that these partitions correspond to D^s and N^s, respectively.

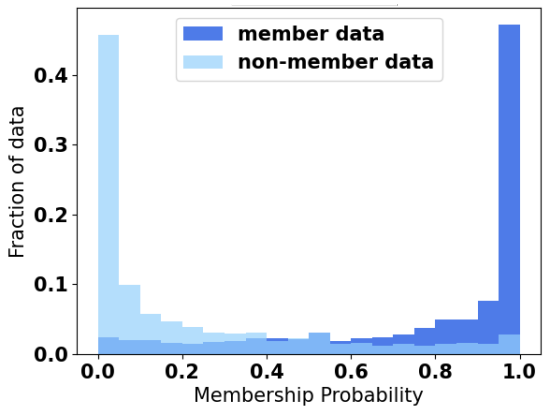


Figure 6.3: Distribution of the membership probability for *members* versus *non-members* data

observe that the users that actually participated in FL are associated with an average higher MP value than those that did not participate. Hence, in this scenario, the MIA attack potentially reveals if a specific user participated to FL.

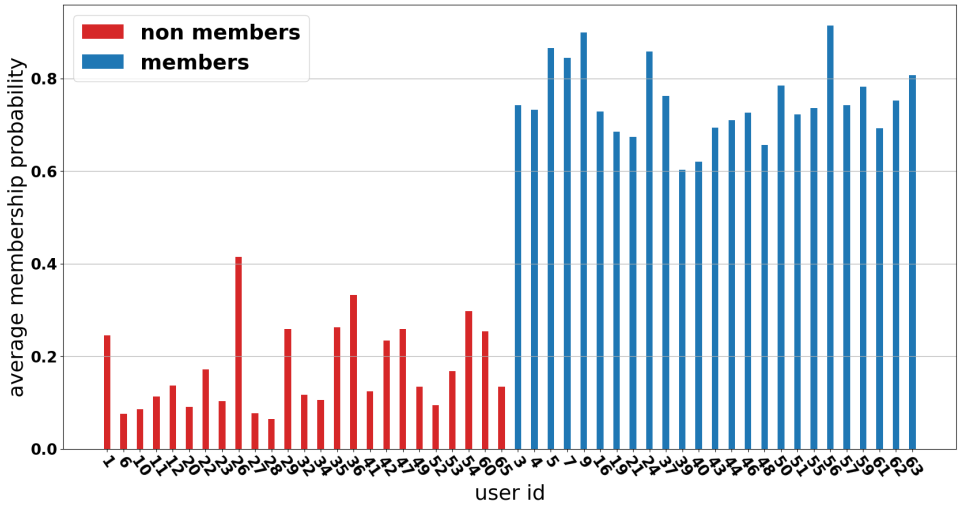


Figure 6.4: Average membership probability assigned by the attack model to each of the considered users.

6.4.3 Evaluating user membership with data not used in FL

In this experiment, we want to check if the attack recognizes the membership of a user even by analyzing data samples from that user that have not been used in training the global model.

Data preparation

In order to perform this experiment, we consider the specific setting where the attacker has access to 15% of data samples (not used to train the global model) from 5% of the FL members. The data partitioning schema is depicted in Figure 6.5.

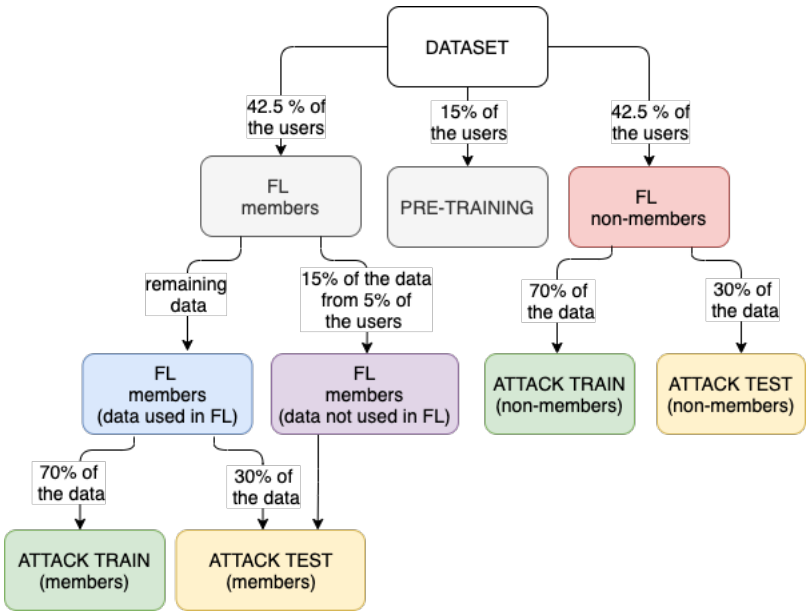


Figure 6.5: Dataset splitting process adopted to evaluate user membership with data not used in FL

Results

Figure 6.6 summarizes the results of the attack at the user granularity. We observed that data samples not used in the FL training still reliably reveal the membership of the corresponding users.

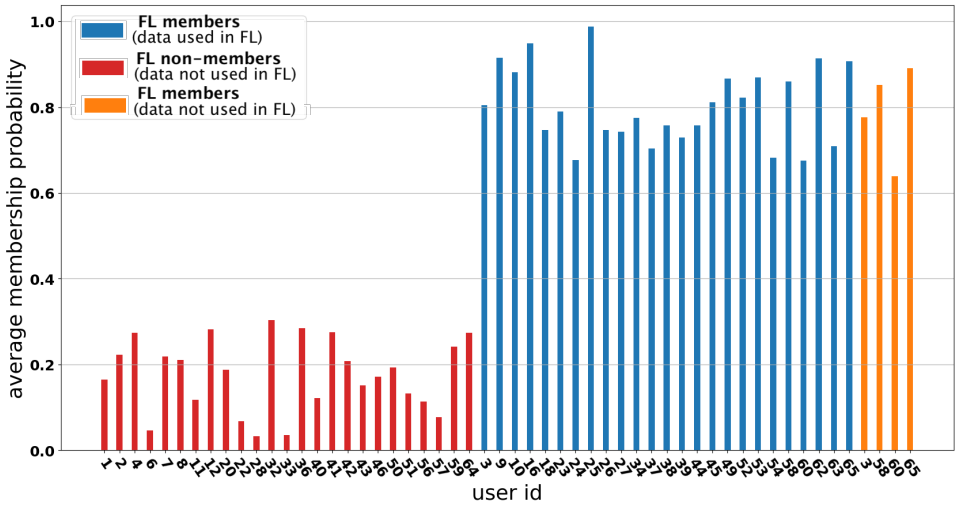
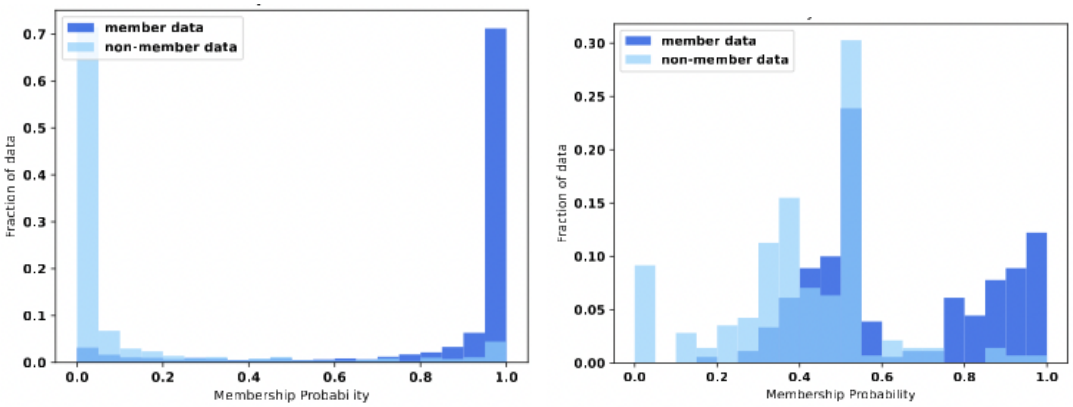


Figure 6.6: Average membership probability assigned by the attack model to each of the considered users.

6.4.4 Evaluating activity membership

In this experiment, we consider the setting proposed in Section 6.4.2 to understand if it is possible to determine whether a user contributed to FL with a specific activity. For each activity, we computed the MP value for each test data sample of both FL members and non-members subjects. Figure 6.7 shows the outcome of this experiment considering the activities *walking* and *sitting*.



(a) MP Walking (b) MP Sitting

Figure 6.7: MP assigned to the samples of the activities *Walking* and *Sitting*

We observe that the attack is effective for *walking* (a) while not really for *sitting* (b). Indeed, Figure 6.7b shows that the average MP is around 0.5. Since both FL members and non-members perform this activity in a similar way, during the training phase the attack model can not observe significant differences in the shadow model behavior when processing members and non-members data.

Intuitively, considering the sensor setup used in MobiAct and in several other HAR datasets, *walking* represents a set of activities that are likely to differ in their pattern of execution by different subjects, while *sitting* represents activities that have limited variance in their execution patterns.

This may lead concluding that the attack for this last category of activities is not effective while it is effective for those in the first category.

Nonetheless, activities in this first category are not necessarily exposed to privacy risks in general. Indeed, considering larger datasets where it is very unlikely that users perform the activity in a unique way, it is questionable if the attack would be effective as well.

Considering possible privacy protection approaches, we believe that these results may provide useful information on which activities may be more exposed, hence guiding, for example, the distribution of artificial noise in obfuscation strategies.

6.5 Summary

In this Chapter, we addressed the research question **Q4)** presented in Section 2.4, by proposing a novel framework based on the Membership Inference Attack that enables evaluating which sensitive information could be inferred by a potential attacker that accesses the model parameters shared in an FL-based system for HAR. Our preliminary results suggest that the global activity model may actually reveal some sensitive information about the participating users. The major limitation of this work was to use of a subset of the target data to learn the attack model (i.e., the worst-case scenario). Obviously, this is not a realistic assumption since the attacker cannot actually access this information. However, we consider the study presented in this Chapter only the first step in a research direction that we intend to explore in the near future.

Chapter 7

Conclusions

7.1 Summary

In this thesis, we proposed novel methodologies to address the limitations related to scalability, privacy, poor personalization, and labeled data scarcity, that emerge in state-of-the-art collaborative approaches for sensor-based HAR. First of all, we introduced an innovative technique that combines context-aware reasoning with semi-supervised learning to increase the number of recognizable activities and reduce the data annotation burden. Nonetheless, this approach keeps suffering from scalability and privacy issues as it involves centralizing the data collected by multiple users into a central server, where the global activity recognition model is trained. Thus, bearing in mind the labeled data scarcity problem, and with the aim of mitigating privacy and scalability issues, we introduced the first hybrid semi-supervised and federated learning (FL) system for HAR. Despite this system allowed us to make an important step towards reducing the above-mentioned issues, it does not overcome the non-IID data concern. Indeed, the FL framework has been designed to perform well with independently and identically distributed data. However, in the field of HAR, that assumption cannot be always satisfied as each subject execute activities in a different way due to its specific physical traits and habits. Therefore, we proposed a Federated Clustering approach for HAR that mitigates the non-IID problem by assigning a personalized classifier for each group of users that exhibit similar ways of performing activities. Finally,

we also considered the possible privacy leakage of FL systems for HAR. In particular, by using a novel membership inference attack methodology we performed a preliminary evaluation on which sensitive information could be inferred by a potential attacker that accesses the model parameters of the global recognition model.

Extensive evaluation with several datasets showed the effectiveness of our methods. In the following, we summarize the specific contributions presented in this thesis.

Hybrid semi-supervised learning and context-aware reasoning framework for collaborative HAR

Our first contribution to this thesis is the hybrid semi-supervised learning and knowledge-based framework that we propose in Chapter 3. Here, a machine learning classifier is in charge of inferring from inertial sensor data the candidate probability distribution over the possible activities. Meanwhile, a knowledge-based reasoning engine is used to refine activity predictions considering context data (i.e., semantic location, weather, moving speed). Precisely, we presented two different typologies of ontology to implement the knowledge-based reasoning engine. The former is a deterministic ontology that uses a rigid ontological formalism to model the relationships between context and activities. The latter is a probabilistic ontology that takes advantage of probabilistic reasoning to capture the intrinsic uncertainty of context data. From our experimental evaluation performed on the DOMINO dataset, emerged that the proposed knowledge-based reasoning engine is effective in both improving the recognition rate of the statistical classifier, and reducing the number of active learning queries used to collect annotated training samples (especially by using the probabilistic ontology). Among the limitations of this work, it is important to note that we did not consider the scalability and privacy issues that may arise in collaboratively training the machine learning model when the number of users increases. Moreover, we aim to study personalization aspects related to personal context situations. Indeed, we believe that incrementally adapting the ontology to each user would allow our system to learn personalized contexts and hence improve accuracy.

FL-based approaches to reduce the data scarcity problem of HAR
Given the scalability and privacy issues that arise in collaboratively training a recognition model for HAR in a labeled data scarcity scenario, in Chapters 4 and Chapter 5 we proposed novel semi-supervised and FL-based methodologies. In particular, in Chapter 4 we introduced FedAR, the first hybrid semi-supervised and federated learning system for HAR. On the one hand, the FL framework enables distributing the training of a global activity recognition model over multiple users in a scalable and privacy-preserving way. On the other hand, a semi-supervised approach based on active learning and label propagating allows to semi-automatically annotate training data by triggering a very limited number of activity queries to the users. Our results showed that this novel method leads to a classification rate that is comparable to the one obtained by HAR methods that rely on fully supervised learning to train the local models.

Despite, the very promising results obtained by FedAR, a limitation of this approach consists of the non-IID data concern. Indeed, users having different physical traits may execute activities with dissimilar motion patterns, hence collecting non-independently and identically distributed datasets. In order to overcome this personalization problem, in Chapter 5, we proposed SS-FedClar, a novel semi-supervised Federated Clustering algorithm that enables assigning a specialized classifier to each group of users who perform activities in a similar way. Our experimental evaluation with publicly available datasets demonstrated that SS-FedClar outperforms FedAR both increasing the classification rate and further reducing the number of active learning queries for the users. Among the limitation of this work, important aspects to consider before a real-world deployment are the need for evaluation on large scale and the problem of dynamically adapting the generated cluster of users when new subjects join the system (or some participants leave it). We will examine in deep these problems and provide possible future research directions in Section 7.2

Investigation of the potential privacy issues in FL-based HAR
In this thesis, we proposed different FL-based approaches with the aim of increasing the privacy level for the users that collaboratively learn a HAR model.

Indeed, the sensor data typically used in HAR may reveal sensitive information about the participating subjects like their habits or health conditions. Sharing this potentially sensitive information with a third-party server with the scope of collaboratively training a shared activity recognition model may pose many privacy threats to the users. In FL each user locally trains a personal model using the available labeled data and sends the updated model parameters to a cloud server that is in charge of aggregating them to generate the shared recognition model. In that way, users preserve the raw sensor data in the local storage of their devices. However, recent studies indicated that even the model's parameters received and manipulated by the cloud server may still reveal sensitive information about the users who participate in FL. However, to the best of our knowledge, non of them focused on the HAR domain.

Thus, we took the first contribution to this line of research by introducing a novel framework to quantitatively evaluate the effectiveness of the Membership Inference Attack (MIA) for FL-based HAR.

Although we used a standard FL setup in our experiments, we gained valuable insights that should be considered when developing FL-based approaches for HAR. For example, we found that even a shared activity recognition model trained with FL may reveal sensitive information about participating users under certain assumptions. However, this study is only a preliminary step, and we plan to explore this research direction further. We acknowledge that using a subset of the target data to learn the attack model is not realistic, and we plan to investigate alternative strategies, such as using GAN to generate synthetic data as well as unsupervised membership attack methods [27, 157]. We also intend to evaluate other types of attacks besides MIA. For instance, the reconstruction attack may be used to recreate sensor patterns that reveal the medical conditions of the participating users, while the property-inference attack could be used to infer high-level properties about specific users from the global activity model. Considering privacy-preserving techniques, we plan to study solutions based on Local Differential Privacy (LDP) with heuristics guided by the outcomes of our analysis. Lastly, we plan to assess the implications of our proposed semi-supervised and Federated Clustering solutions on the privacy level of the users.

7.2 Future Works

The results presented in this thesis are encouraging, and we plan to further improve our methods by investigating several interesting research directions. In the following, we outline the ones we believe are more promising.

Acceptability of active learning based strategies for HAR
In this thesis, we proposed various approaches that use active learning in order to obtain annotated samples from the users. However, we assumed that users are willing to provide active learning feedback if the number of queries will quickly decreases over time. This assumption is shared with other HAR papers based on active learning [14, 15]. Nevertheless, it may not be realistic in the HAR domain, since users' availability is highly influenced by the contexts in which queries are received. For instance, a user may not be willing to provide feedback while participating in a social event. Postponing queries is critical since it becomes challenging to locate in time and remember the activity that was performed. Indeed, each active learning query is associated with a single feature vector processed by the classifier at a specific time instant.

Moreover, another challenge that we encountered when using user-provided labels is the possibility of wrong labels, either due to human error or misunderstanding. This issue is not unique to our approach but is a general problem that affects all supervised learning approaches that rely on user-provided labels. There are several solutions to deal with wrongly labeled data, including filtering out the incorrect labels, relabeling the data, or using robust learning techniques.

One possible approach to filtering out incorrect labels is to use a validation set that is separate from the training data. The validation set can be used to detect and remove samples with inconsistent or incorrect labels. Another approach is to use algorithms that are robust to mislabeled data, such as the Co-Training or Tri-Training algorithms [158], which use multiple classifiers trained on different subsets of the data to identify and correct mislabeled samples.

Relabeling the data can also be an effective solution to the issue of wrong labels. This can be done by asking the user to review and correct their previously provided labels or by using crowd-sourcing platforms to obtain new labels from

a large group of users. However, this approach may not always be feasible due to the cost and time required to obtain new labels.

Finally, robust learning techniques can be used to mitigate the effect of wrong labels. For example, some techniques use robust loss functions, such as the Huber loss function, that are less sensitive to outliers and mislabeled data [159].

Currently, we are exploring the use of these solutions to deal with wrongly labeled data in the context of our active learning-based HAR approach. We are also investigating the feasibility of combining these solutions with our active learning module to obtain more accurate labels from the users.

Include context data into federate learning

In chapter 3 we introduced a novel semi-supervised learning and context-aware reasoning framework for collaborative HAR. From our experimental evaluation emerged that the proposed context-based refinement is effective in both improving the recognition rate, and reducing the number of active learning queries used to collect annotated training samples. Then, in Chapters 4 and 5, we proposed other semi-supervised approaches for HAR that, by leveraging the FL framework, achieve the objective of reducing the users' burden in annotating activity examples, while improving the system scalability and providing more privacy guarantees for the involved subjects. However, these FL-based approaches did not take advantage of context data in the activity recognition process. Therefore, in the near future, we intend to investigate how to include high-level context data to continuously adapt the federated model based on the current user's context. For instance, if the user is in the gym, she could use and improve a model that is specific for physical exercises. On the other hand, if she is at home, the system would consider a federated model more suitable for smart-home environments.

Evaluation on a large scale

In this thesis, we evaluated our collaborative learning approaches for HAR by relying on those public datasets with the highest number of subjects. However, real-world scenarios may involve thousands or even millions of users. Hence, although the proposed methodologies exhibited promising results, they need to be confirmed in more realistic experiments on a larger scale. For instance, consid-

ering the Federated learning approaches proposed in Chapters 4 and Chapter 5, a significant limitation is that at each communication round every participating client is involved in the global model update. However, for the sake of scalability, real-word FL methods randomly sample a limited number of clients at each communication round [19]. Hence, we will investigate scalable solutions to distribute the process in multiple communication rounds. Another significant problem related to deploying our FL-based solutions on a large scale is the correct choice of hyper-parameters. Indeed, the hyper-parameters that proved to be effective in our experiments may not reflect the ones that are effective on a large scale. Hence, we will study the challenging problem of choosing the correct hyper-parameters in large-scale scenarios, where only a limited amount of labeled data is actually available.

Continual federated clustering

In Chapter 5, we evaluated our novel semi-supervised and federated clustering approach by using publicly available datasets that only include a limited number of data samples for each user. In a real-world scenario, users may change their activity patterns and habits in the long term. Moreover, considering a real-world deployment, the set of clients may significantly vary due to new clients that join the system as well as clients that leave the system. When those events occur, the clustering structure may change. In order to tackle this challenge, a possibility is that the cloud server stores every intermediate model computed during clustering (i.e., the dendograms associated with intermediate steps of hierarchical clustering). Hence, when needed, the server can recompute an optimal set of clusters by reversing some of the clustering steps and evaluating the new situation. A similar approach was proposed in [111]. However, re-computing clusters may be computationally expensive, and it should be performed periodically only when the above-mentioned conditions change significantly. For instance, considering changes in local data distributions, each client may locally run algorithms to detect significant changes in patterns and habits. When a client detects such a change, it may notify the server. The clustering update may be performed only when a significant number of clients have notified the server. Given the changes

in the set of users, a clustering update may be required only when there is a significant amount of join and abandon events. In the future, we intend to explore in deep these problems and propose a valuable and efficient solution to tackle them.

Self-Supervised learning

In this thesis, we presented different semi-supervised learning based approaches that enable mitigating the data scarcity problem in collaborative HAR. However, all of those solutions involve initializing the recognition model with a small portion of human-annotated data, and then incrementally training it thanks to an active-learning strategy. However, by following this approach may arise some issues. Indeed, as we previously mentioned, users characterized by different physical traits may perform activities in different ways, and a recognition model initialized with few examples would probably struggle in capturing those differences. Accordingly, some users would probably receive a considerable number of active-learning queries, especially in the early stages. Therefore, it emerges the need for a model initialized over the labeled examples collected by heterogeneous users. However, as we extensively discussed in this thesis, this is often unfeasible on large scale due to the activity data annotation costs. A valuable solution to tackle this problem may consist of leveraging self-supervised learning (SSL) as it provides a general and powerful framework for learning with a tremendous amount of unlabeled inputs through solving pretext tasks [65]. Overall, in SSL a surrogate objective (i.e., the pretext task) is specified in such a way that optimizing it would force the network to learn meaningful and usable features for the downstream task (i.e., classification). Then, only a few labeled data are sufficient to refine the network for classification. Thus, in the near future, we aim to exploit SSL to initialize the recognition model over a very large number of unlabeled data coming from plenty of subjects. Then, a specifically designed active learning strategy can be used to fine-tune it over each user with a very limited set of labeled examples.

Acknowledgement

I wish to express my deepest gratitude to my supervisor Claudio Bettini for his unwavering support and guidance throughout my doctoral studies. His wise counsel and expertise were instrumental in my success.

I extend my special thanks to my co-supervisor Gabriele Civitarese, whose careful mentorship and support helped me navigate the intricate world of research in my early stages.

I am also grateful to Philippe Lalanda for generously hosting me at the University of Grenoble for a successful research visit, and to Sannara Ek for the great and enjoyable experience of collaborating on challenging problems.

I would like to acknowledge Philippe Lalanda, Franca Delmastro, and Nirmalya Roy for their valuable comments and suggestions, which significantly improved the quality of this thesis.

Heartfelt thanks go to the students who contributed their expertise and diligence to this work: Luigi Tropiano, Emanuela Elli, Vincenzo Raffa, Giorgia Masoero, and Carolin Ghali.

I am deeply grateful to Luca Arrotta for his support and companionship throughout this journey, as well as to other members of the EWLab for sharing enjoyable moments with me.

I want to express my gratitude to my family for never stopping believing in me, and to my friends for their unwavering support.

Lastly, I want to express my love and gratitude to Francesca, whose constant presence in my life has made it easier and more beautiful.

Bibliography

[1] O. D. Lara, M. A. Labrador, *et al.*, "A survey on human activity recognition using wearable sensors.," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[2] F. Demrozi, G. Pravadelli, A. Bihorac, and P. Rashidi, "Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey," *IEEE Access*, vol. 8, pp. 210816–210836, 2020.

[3] A. D. Antar, M. Ahmed, and M. A. R. Ahad, "Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: a review," in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pp. 134–139, IEEE, 2019.

[4] N. Roy, A. Misra, and D. Cook, "Infrastructure-assisted smartphone-based adl recognition in multi-inhabitant smart environments," in *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 38–46, IEEE, 2013.

[5] F. Rustam, A. A. Reshi, I. Ashraf, A. Mehmood, S. Ullah, D. M. Khan, and G. S. Choi, "Sensor-based human activity recognition using deep stacked multilayered perceptron model," *IEEE Access*, vol. 8, pp. 218898–218910, 2020.

[6] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Trans. Syst. Man Cybern. C:App. Rev.*, vol. 42, no. 6, pp. 790–808, 2012.

[7] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, 2012.

[8] S.-M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," in *2017 ieee international conference on big data and smart computing (bigcomp)*, pp. 131–134, IEEE, 2017.

[9] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing: Second International Conference, PERVASIVE 2004,*

Linz/Vienna, Austria, April 21-23, 2004. Proceedings, (Berlin, Heidelberg), pp. 1–17, Springer, 2004.

[10] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Procedia Computer Science*, vol. 155, pp. 698–703, 2019.

[11] C. Bettini and D. Riboni, "Privacy protection in pervasive systems: State of the art and technical challenges," *Pervasive Mob. Comput.*, vol. 17, pp. 159–174, 2015.

[12] D. J. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowl. Inf. Sys.*, vol. 36, no. 3, pp. 537–556, 2013.

[13] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Activity recognition with evolving data streams: A review," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, p. 71, 2018.

[14] R. Smith and M. Dragone, "A dialogue-based interface for active learning of activities of daily living," in *27th International Conference on Intelligent User Interfaces*, pp. 820–831, 2022.

[15] H. S. Hossain, M. A. A. H. Khan, and N. Roy, "Active learning enabled activity recognition," *Pervasive and Mobile Computing*, vol. 38, pp. 312–330, 2017.

[16] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Adaptive mobile activity recognition system with evolving data streams," *Neurocomputing*, vol. 150, pp. 304–317, 2015.

[17] L. Liao, D. Fox, and H. Kautz, "Location-based activity recognition," in *Advances in Neural Information Processing Systems*, pp. 787–794, 2006.

[18] D. Riboni and C. Bettini, "COSAR: Hybrid reasoning for context-aware activity recognition," *Personal and Ubiquitous Computing*, vol. 15, no. 3, pp. 271–289, 2011.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.

[20] S. Ek, F. Portet, P. Lalanda, and G. Vega, "Evaluation of federated learning aggregation algorithms: application to human activity recognition," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pp. 638–643, 2020.

[21] K. Sozinov, V. Vlassov, and S. Girdzijauskas, "Human activity recognition using federated learning," in *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications*, pp. 1103–1111, IEEE, 2018.

[22] Z. Xiao, X. Xu, H. Xing, F. Song, X. Wang, and B. Zhao, "A federated learning system with enhanced feature extraction for human activity recognition," *Knowledge-Based Systems*, vol. 229, p. 107338, 2021.

[23] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, 2020.

[24] G. M. Weiss and J. Lockhart, "The impact of personalization on smartphone-based activity recognition," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, Citeseer, 2012.

[25] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, "Clusterfl: a similarity-aware federated learning system for human activity recognition," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 54–66, 2021.

[26] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2512–2520, IEEE, 2019.

[27] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 739–753, 2019.

[28] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017.

[29] C. Bettini, G. Civitarese, and R. Presotto, "Caviar: Context-driven active and incremental activity recognition," *Knowledge-Based Systems*, vol. 196, p. 105816, 2020.

[30] C. Bettini, G. Civitarese, D. Giancane, and R. Presotto, "Procaviar: Hybrid data-driven and probabilistic knowledge-based activity recognition," *IEEE Access*, vol. 8, pp. 146876–146886, 2020.

[31] L. Cao, Y. Wang, B. Zhang, Q. Jin, and A. V. Vasilakos, "Gchar: An efficient group-based context—aware human activity recognition on smartphone," *Journal of Parallel and Distributed Computing*, vol. 118, pp. 67–80, 2018.

[32] F. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[33] R. M. Cichy, A. Khosla, D. Pantazis, A. Torralba, and A. Oliva, "Deep neural networks predict hierarchical spatio-temporal cortical dynamics of human visual object recognition," *arXiv preprint arXiv:1601.02970*, 2016.

[34] L. Onofri, P. Soda, M. Pechenizkiy, and G. Iannello, "A survey on using domain and contextual knowledge for human activity recognition in video streams," *Expert Systems with Applications*, vol. 63, pp. 97–111, 2016.

[35] S. Savazzi, V. Rampa, F. Vicentini, and M. Giussani, "Device-free human sensing and localization in collaborative human–robot workspaces: A case study," *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1253–1264, 2015.

[36] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," *IEEE Sensors Journal*, vol. 17, no. 2, pp. 386–403, 2016.

[37] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recogn. Lett.*, vol. 119, pp. 3–11, 2019.

[38] M. F. A. bin Abdullah, A. F. P. Negara, M. S. Sayeed, D.-J. Choi, and K. S. Muthu, "Classification algorithms in human activity recognition using smartphones," *International Journal of Computer and Information Engineering*, vol. 6, no. 77-84, p. 106, 2012.

[39] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Iberian conference on pattern recognition and image analysis*, pp. 289–296, Springer, 2011.

[40] Z. Feng, L. Mo, and M. Li, "A random forest-based ensemble method for activity recognition," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5074–5077, IEEE, 2015.

[41] D. N. Tran and D. D. Phan, "Human activities recognition in android smartphone using support vector machine," in *2016 7th international conference on intelligent systems, modelling and simulation (isms)*, pp. 64–68, IEEE, 2016.

[42] K. M. Chathuramali and R. Rodrigo, "Faster human activity recognition with svm," in *International conference on advances in ICT for emerging regions (ICTer2012)*, pp. 197–203, IEEE, 2012.

[43] S. Sani, N. Wiratunga, S. Massie, and K. Cooper, "knn sampling for personalised human activity recognition," in *International conference on case-based reasoning*, pp. 330–344, Springer, 2017.

[44] M. Kose, O. D. Incel, and C. Ersoy, "Online human activity recognition on smart phones," in *Workshop on mobile sensing: from smartphones and wearables to big data*, vol. 16, pp. 11–15, 2012.

[45] P. Asghari, E. Soleimani, and E. Nazerfard, "Online human activity recognition employing hierarchical hidden markov models," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1141–1152, 2020.

[46] A. Jalal, S. Kamal, and D. Kim, "Human depth sensors-based activity recognition using spatiotemporal features and hidden markov model for smart environments," *Journal of computer networks and communications*, vol. 2016, 2016.

[47] D. Figo, P. C. Diniz, D. R. Ferreira, and J. M. Cardoso, "Preprocessing techniques for context recognition from accelerometer data," *Personal and Ubiquitous Computing*, vol. 14, no. 7, pp. 645–662, 2010.

[48] M. Panwar, S. R. Dyuthi, K. C. Prakash, D. Biswas, A. Acharyya, K. Maharatna, A. Gautam, and G. R. Naik, "Cnn based approach for activity recognition using a wrist-worn accelerometer," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2438–2441, IEEE, 2017.

[49] Y. Guan and T. Plötz, "Ensembles of deep lstm learners for activity recognition using wearables," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–28, 2017.

[50] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*, pp. 108–109, IEEE, 2012.

[51] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[52] N. Györbíró, Á. Fábián, and G. Hományi, "An activity recognition system for mobile phones," *Mobile Networks and Applications*, vol. 14, no. 1, pp. 82–91, 2009.

[53] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," in *International conference on ubiquitous intelligence and computing*, pp. 548–562, Springer, 2010.

[54] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys*, vol. 46, no. 3, pp. 33:1–33:33, 2014.

[55] Y. Kwon, K. Kang, and C. Bae, "Unsupervised learning for human activity recognition using smartphone sensors," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6067–6074, 2014.

[56] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[57] K. M. Rashid and J. Louis, "Times-series data augmentation and deep learning for construction equipment activity recognition," *Advanced Engineering Informatics*, vol. 42, p. 100944, 2019.

[58] J. Wang, Y. Chen, Y. Gu, Y. Xiao, and H. Pan, "Sensorygans: An effective generative adversarial framework for sensor-based human activity recognition," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018.

[59] M. H. Chan and M. H. M. Noor, "A unified generative model using generative adversarial network for activity recognition," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–10, 2020.

[60] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowl. Inf. Sys.*, vol. 36, no. 3, pp. 537–556, 2013.

[61] J. Wang, V. W. Zheng, Y. Chen, and M. Huang, "Deep transfer learning for cross-domain activity recognition," in *proceedings of the 3rd International Conference on Crowd Science and Engineering*, pp. 1–8, 2018.

[62] A. R. Sanabria, F. Zambonelli, and J. Ye, "Unsupervised domain adaptation in activity recognition: A gan-based approach," *IEEE Access*, vol. 9, pp. 19421–19438, 2021.

[63] E. Soleimani and E. Nazerfard, "Cross-subject transfer learning in human activity recognition systems using generative adversarial networks," *Neurocomputing*, vol. 426, pp. 26–34, 2021.

[64] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[65] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[66] C. I. Tang, I. Perez-Pozuelo, D. Spathis, S. Brage, N. Wareham, and C. Mascolo, "Selfhar: Improving human activity recognition through self-training with unlabeled data," *arXiv preprint arXiv:2102.06073*, 2021.

[67] H. Haresamudram, I. Essa, and T. Plötz, "Assessing the state of self-supervised human activity recognition using wearables," *arXiv preprint arXiv:2202.12938*, 2022.

[68] B. Longstaff, S. Reddy, and D. Estrin, "Improving activity classification for health applications on mobile devices using active and semi-supervised learning," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, pp. 1–7, IEEE, 2010.

[69] M. Stikic, K. Van Laerhoven, and B. Schiele, "Exploring semi-supervised and active learning for activity recognition," in *2008 12th IEEE International Symposium on Wearable Computers*, pp. 81–88, IEEE, 2008.

[70] D. Guan, W. Yuan, Y.-K. Lee, A. Gavrilov, and S. Lee, "Activity recognition based on semi-supervised learning," in *Embedded and Real-Time Computing Systems and Applications, 2007. RTCSA 2007. 13th IEEE International Conference on*, pp. 469–475, IEEE, 2007.

[71] M. Stikic, D. Larlus, and B. Schiele, "Multi-graph based semi-supervised learning for activity recognition," in *2009 international symposium on wearable computers*, pp. 85–92, IEEE, 2009.

[72] Y.-S. Lee and S.-B. Cho, "Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data," *Neurocomputing*, vol. 126, pp. 106–115, 2014.

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

[74] K. T. Nguyen, F. Portet, and C. Garbay, "Dealing with imbalanced data sets for human activity recognition using mobile phone sensors," in *3rd International Workshop on Smart Sensing Systems*, 2018.

[75] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, 2010.

[76] D. Riboni and C. Bettini, "Owl 2 modeling and reasoning with complex human activities," *Pervasive and Mobile Computing*, vol. 7, no. 3, pp. 379–395, 2011.

[77] F. Baader, D. Calvanese, D. McGuinness, P. Patel-Schneider, D. Nardi, *et al.*, *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

[78] L. Chen and C. Nugent, "Ontology-based activity recognition in intelligent pervasive environments," *International Journal of Web Information Systems*, 2009.

[79] H. Chen, T. Finin, and A. Joshi, "The soupa ontology for pervasive computing," in *Ontologies for agents: Theory and experiences*, pp. 233–258, Springer, 2005.

[80] G. Meditskos, S. Dasiopoulou, and I. Kompatsiaris, "Metaq: A knowledge-driven framework for context-aware activity recognition combining sparql and owl 2 activity patterns," *Pervasive and Mobile Computing*, vol. 25, pp. 104–124, 2016.

[81] Q. Ni, I. Pau de la Cruz, and A. B. Garcia Hernando, "A foundational ontology-based model for human activity representation in smart homes," *Journal of Ambient Intelligence and Smart Environments*, vol. 8, no. 1, pp. 47–61, 2016.

[82] O. Banos, C. Villalonga, J. Bang, T. Hur, D. Kang, S. Park, V. Le-Ba, M. B. Amin, M. A. Razzaq, W. A. Khan, *et al.*, "Human behavior analysis by means of multimodal context mining," *Sensors*, vol. 16, no. 8, p. 1264, 2016.

[83] P. Patel, A. Gyrard, D. Thakker, A. P. Sheth, and M. Serrano, "Swotsuite: A toolkit for prototyping cross-domain semantic web of things applications.," in *International Semantic Web Conference (Posters & Demos)*, 2016.

[84] A. Ranganathan, J. Al-Muhtadi, and R. H. Campbell, "Reasoning about uncertain contexts in pervasive computing environments," *IEEE Pervasive computing*, vol. 3, no. 2, pp. 62–70, 2004.

[85] G. Stoilos, G. Stamou, J. Z. Pan, V. Tzouvaras, and I. Horrocks, "Reasoning with very expressive fuzzy description logics," *Journal of Artificial Intelligence Research*, vol. 30, pp. 273–320, 2007.

[86] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1, pp. 107–136, 2006.

[87] G. Civitarese, T. Sztyler, D. Riboni, C. Bettini, and H. Stuckenschmidt, "Polaris: Probabilistic and ontological activity recognition in smart-homes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 1, pp. 209–223, 2019.

[88] R. N. Carvalho, K. B. Laskey, and P. C. Costa, "Pr-owl–a language for defining probabilistic ontologies," *International Journal of Approximate Reasoning*, vol. 91, pp. 56–79, 2017.

[89] L. L. dos Santos, R. N. Carvalho, M. Ladeira, L. Weigang, and G. L. Mendes, "Pr-owl 2 rl-a language for scalable uncertainty reasoning on the semantic web," in *11th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2015)*, p. 14, 2015.

[90] E. Bellodi, E. Lamma, F. Riguzzi, S. Albani, *et al.*, "A distribution semantics for probabilistic ontologies.," *URSW*, vol. 778, pp. 75–86, 2011.

[91] F. Riguzzi, E. Bellodi, E. Lamma, and R. Zese, "Probabilistic description logics under the distribution semantics," *Semantic Web*, vol. 6, no. 5, pp. 477–501, 2015.

[92] M. Niepert, J. Noessner, and H. Stuckenschmidt, "Log-linear description logics," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2153–2158, 2011.

[93] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[94] G. Civitarese, J. Ye, M. Zampatti, and C. Bettini, "Collaborative activity recognition with heterogeneous activity sets and privacy preferences," *Journal of Ambient Intelligence and Smart Environments*, no. Preprint, pp. 1–20, 2021.

[95] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[96] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[97] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

[98] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[99] G. Damaskinos, R. Guerraoui, A.-M. Kermarrec, V. Nitu, R. Patra, and F. Taiani, "Fleet: Online federated learning via staleness awareness and performance prediction," in *Proceedings of the 21st International Middleware Conference*, pp. 163–177, 2020.

[100] S. Ek, F. Portet, P. Lalanda, and G. Vega, "A federated learning aggregation algorithm for pervasive computing: Evaluation and comparison," in *19th IEEE International Conference on Pervasive Computing and Communications PerCom 2021*, 2021.

[101] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent iot applications: A cloud-edge based framework," *IEEE Computer Graphics and Applications*, 2020.

[102] Y. Zhao, H. Liu, H. Li, P. Barnaghi, and H. Haddadi, "Semi-supervised federated learning for activity recognition," *arXiv preprint arXiv:2011.00851*, 2020.

[103] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, 2020.

[104] V. Kelli, V. Argyriou, T. Lagkas, G. Fragulis, E. Grigoriou, and P. Sarigiannidis, "Ids for industrial applications: a federated learning approach with active personalization," *Sensors*, vol. 21, no. 20, p. 6743, 2021.

[105] A. Saeed, F. D. Salim, T. Ozcelebi, and J. Lukkien, "Federated self-supervised learning of multisensor representations for embedded intelligence," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1030–1040, 2020.

[106] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[107] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Adv. Neural Inf. Process. Syst.*, pp. 3320–3328, 2014.

[108] T. Yu, T. Li, Y. Sun, S. Nanda, V. Smith, V. Sekar, and S. Seshan, "Learning context-aware policies from multiple smart homes via federated multi-task learning," in *2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 104–115, IEEE, 2020.

[109] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, IEEE, 2020.

[110] Z. Chen, P. Tian, W. Liao, and W. Yu, "Zero knowledge clustering based adversarial mitigation in heterogeneous federated learning," *IEEE Trans. Netw. Sci. Eng.*, 2020.

[111] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020.

[112] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE Transactions on Services Computing*, pp. 1–1, 2019.

[113] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 691–706, 2019.

[114] B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the gan: Information leakage from collaborative deep learning," *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[115] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," pp. 250–258, 06 2020.

[116] L. Zhu, Z. Liu, and S. Han, *Deep Leakage from Gradients*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[117] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, (New York, NY, USA), p. 1322–1333, Association for Computing Machinery, 2015.

[118] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2018.

[119] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang, "Privacy at scale: Local differential privacy in practice," in *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, (New York, NY, USA), p. 1655–1658, Association for Computing Machinery, 2018.

[120] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, *Practical Secure Aggregation for Privacy-Preserving Machine Learning*, p. 1175–1191. New York, NY, USA: Association for Computing Machinery, 2017.

[121] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The mobiact dataset: Recognition of activities of daily living using smartphones.," in *ICT4AgeingWell*, pp. 143–151, 2016.

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

[123] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, pp. 437–442, 2013.

[124] B. Almaslukh, J. AlMuhtadi, and A. Artoli, "An effective deep autoencoder approach for online smartphone-based human activity recognition," *Int. J. Comput. Sci. Netw. Secur*, vol. 17, no. 4, pp. 160–165, 2017.

[125] T. Plötz, N. Y. Hammerla, and P. L. Olivier, "Feature learning for activity recognition in ubiquitous computing," in *Twenty-second international joint conference on artificial intelligence*, 2011.

[126] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, "A survey of context modelling and reasoning techniques," *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, 2010.

[127] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.

[128] I. Guyon and A. Elisseeff, "An introduction to feature extraction," in *Feature extraction*, pp. 1–25, Springer, 2006.

[129] J. Noessner and M. Niepert, "Elog: a probabilistic reasoner for owl el," in *Proceedings of the 5th international conference on Web reasoning and rule systems*, pp. 281–286, Springer, 2011.

[130] R. Helaoui, D. Riboni, and H. Stuckenschmidt, "A probabilistic ontological framework for the recognition of multilevel human activities," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, (New York, NY, USA), pp. 345–354, ACM, 2013.

[131] G. Civitarese, C. Bettini, T. Sztyler, D. Riboni, and H. Stuckenschmidt, "newnectar: Collaborative active learning for knowledge-based probabilistic activity recognition," *Pervasive and Mobile Computing*, vol. 56, pp. 88–105, 2019.

[132] I. Žliobaitė, A. Bifet, B. Pfahringer, and G. Holmes, "Active learning with drifting streaming data," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 1, pp. 27–39, 2013.

[133] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 1393–1400, IEEE, 2009.

[134] T. Sztyler and H. Stuckenschmidt, "Online personalization of cross-subjects based activity recognition models on wearable devices," in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 180–189, IEEE, 2017.

[135] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang, "Hermit: an owl 2 reasoner," *Journal of Automated Reasoning*, vol. 53, no. 3, pp. 245–269, 2014.

[136] M. Horridge and S. Bechhofer, "The owl api: A java api for owl ontologies," *Semantic Web*, vol. 2, no. 1, pp. 11–21, 2011.

[137] J. Gama, R. Sebastião, and P. P. Rodrigues, "On evaluating stream learning algorithms," *Machine learning*, vol. 90, no. 3, pp. 317–346, 2013.

[138] P. Palmes, H. K. Pung, T. Gu, W. Xue, and S. Chen, "Object relevance weight pattern mining for activity recognition and segmentation," *Pervasive and Mobile Computing*, vol. 6, no. 1, pp. 43–57, 2010.

[139] D. Riboni and M. Murtas, "Sensor-based activity recognition: One picture is worth a thousand words," *Future Generation Computer Systems*, vol. 101, pp. 709–722, 2019.

[140] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[141] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753, IEEE, 2019.

[142] F. Cruciani, A. Vafeiadis, C. Nugent, I. Cleland, P. McCullagh, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, and R. Hamzaoui, "Comparing cnn and human crafted features for human activity recognition," in *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 960–967, IEEE, 2019.

[143] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in neural information processing systems*, vol. 16, no. 16, pp. 321–328, 2004.

[144] N. Widmann and S. Verberne, "Graph-based semi-supervised learning for text classification," in *Proceedings of the ACM SIGIR international conference on theory of information retrieval*, pp. 59–66, 2017.

[145] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[146] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 743–755, 2020.

[147] R. Presotto, G. Civitarese, and C. Bettini, "Semi-supervised and personalized federated activity recognition based on active learning and label propagation," *Personal and Ubiquitous Computing*, pp. 1–18, 2022.

[148] R. Presotto, G. Civitarese, and C. Bettini, "Fedclar: Federated clustering for personalized sensor-based human activity recognition," in *2022 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 227–236, IEEE, 2022.

[149] W. Mou, C. Fu, Y. Lei, and C. Hu, "A verifiable federated learning scheme based on secure multi-party computation," in *International Conference on Wireless Algorithms, Systems, and Applications*, pp. 198–209, Springer, 2021.

[150] D. Zhang, X. Chen, D. Wang, and J. Shi, "A survey on collaborative deep learning and privacy-preserving," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 652–658, IEEE, 2018.

[151] L. Zhu and S. Han, "Deep leakage from gradients," in *Federated learning*, pp. 17–31, Springer, 2020.

[152] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, IEEE, 2017.

[153] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 691–706, IEEE, 2019.

[154] L. Lyu, X. He, Y. W. Law, and M. Palaniswami, "Privacy-preserving collaborative deep learning with application to human activity recognition," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1219–1228, 2017.

[155] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv:1806.01246*, 2018.

[156] C. Chatzaki, M. Pediaditis, G. Vavoulas, and M. Tsiknakis, "Human daily activity and fall recognition using a smartphone's acceleration sensor," pp. 100–118, 07 2017.

[157] Y. Bai, D. Chen, T. Chen, and M. Fan, "Ganmia: Gan-based black-box membership inference attack," in *ICC 2021-IEEE International Conference on Communications*, pp. 1–6, IEEE, 2021.

[158] K. D. Garcia, C. R. de Sá, M. Poel, T. Carvalho, J. Mendes-Moreira, J. M. Cardoso, A. C. de Carvalho, and J. N. Kok, "An ensemble of autonomous auto-encoders for human activity recognition," *Neurocomputing*, vol. 439, pp. 271–280, 2021.

[159] S. Huang, "Robust learning of huber loss under weak conditional moment," *Neurocomputing*, vol. 507, pp. 191–198, 2022.