UNIVERSITÀ DEGLI STUDI DI MILANO

PhD School in
Computer Science

Computer Science Department
"Giovanni Degli Antoni"

PhD in
Computer Science
XXXVI Cycle

Neuro-Symbolic AI Approaches for Sensor-Based Human Activity Recognition

INF/01

PhD candidate:
Luca ARROTTA

Advisor:
Prof. Claudio BETTINI

Co-Advisor:
Dr. Gabriele CIVITARESE

PhD Coordinator:
Prof. Roberto SASSI

Academic Year 2022/2023

Now and then I try to find
A place in my mind
Where you can stay awake
Forever

Abstract

Sensor-based Human Activity Recognition (HAR) is an active research area, with relevant applications in healthcare and well-being. Deep Learning (DL) classifiers are currently the leading approach to tackle HAR, but their deployment is often limited by their inherent opacity and the scarcity of labeled training data. Fortunately, common sense and domain knowledge about activity execution can improve purely data-driven approaches. Indeed, in the general machine learning community, Neuro-Symbolic AI (NeSy) methods are emerging to combine DL models with more traditional symbolic AI techniques that rely on knowledge-based reasoning to improve models' interpretability while reducing their reliance on labeled data during training.

This thesis explores innovative NeSy solutions proposed to enhance sensor-based HAR. The initial chapters focus on NeSy methods designed to mitigate the scarcity of labeled training data. Considering smart-home environments inhabited by multiple subjects, a main problem is *data association*, i.e., correctly associating sensor events (e.g., the opening of the fridge) with the subject(s) that actually generated them. Most works in the literature addressed this challenge with purely data-driven solutions, thus aggravating the labeled data scarcity problem. For this reason, we propose a NeSy method that relies on symbolic reasoning to tackle data association without the need for any labeled data.

Moreover, we also address data scarcity for context-aware HAR based on mobile devices. While NeSy approaches have been already proposed in this research area, they rely on domain knowledge only after the training process of the DL classifier. This limits its ability to handle data uncertainty. Hence, we present two novel NeSy approaches that infuse domain knowledge into DL classifiers during their learning process. Experimental results show how such methods reduce

the amount of labeled data required during training while being more robust to noisy data compared to state-of-the-art NeSy methods.

Finally, we present an initial investigation of interpretability aspects. We introduce a metric that quantitatively evaluates, based on domain knowledge, the quality of explanations obtained from DL activity classifiers. Due to time constraints, this metric has been currently used only to evaluate purely data-driven approaches. Nonetheless, we plan to employ such a metric to quantify the interpretability benefits provided by NeSy methods for HAR.

Overall, all the methods presented in this thesis have been experimentally evaluated on publicly available datasets that have been collected in controlled or in-the-wild settings.

Author's publications

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

Journals

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"Semantic Loss: a new Neuro-Symbolic approach for Context-Aware Human Activity Recognition"*. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2023, to appear.

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"Probabilistic Knowledge Infusion through Symbolic Features for Context-Aware Activity Recognition"*. Pervasive and Mobile Computing, Elsevier, 2023. (DOI: 10.1016/j.pmcj.2023.101780)

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"MICAR: Multi-Inhabitant Context-Aware Activity Recognition in Home Environments"*. Distributed and Parallel Database, Springer, 2022. (DOI: 10.1007/s10619-022-07403-z)

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"DeXAR: Deep Explainable Sensor-Based Activity Recognition in Smart-Home Environments"*. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2022. (DOI: 10.1145/3517224)

International conferences

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"SelfAct: Personalized Activity Recognition based on Self-Supervised and Active Learning"*. In Proceedings of the 20th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous), EAI, 2023, to appear.

- Luca Arrotta, Gabriele Civitarese, Michele Fiori, Claudio Bettini, *"Explaining Human Activities Instances Using Deep Learning Classifiers"*. In 2022 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2022.

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"Knowledge Infusion for Context-Aware Sensor-Based Human Activity Recognition"*. In 2022 IEEE International Conference on Smart Computing (SmartComp), 2022.

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"The MARBLE Dataset: Multi-Inhabitant Activities of Daily Living Combining Wearable and Environmental Sensors Data"*. In International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (MobiQuitous). Cham: Springer International Publishing, 2021.

- 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

- Luca Arrotta, Gabriele Civitarese, Riccardo Presotto, Claudio Bettini, *"DOMINO: A Dataset for Context-Aware Human Activity Recognition using Mobile Devices"*. In 2023 24th IEEE International Conference on Mobile Data Management (MDM) Workshops. IEEE, 2023.

- Luca Arrotta, *"Multi-inhabitant and Explainable Activity Recognition in Smart Homes"*. In 2021 22nd IEEE International Conference on Mobile Data Management (MDM) (pp. 264-266). IEEE, 2021.

Contents

Abstract **3**

Author's publications **5**

1 Introduction **18**

1.1 Motivation 18
1.2 Problem definition 19
1.3 An ideal neuro-symbolic framework for HAR 21
1.4 Research contributions 24
1.4.1 Neuro-symbolic HAR in multi-subject smart-home environments 24
1.4.2 Knowledge infusion through symbolic features for context-aware HAR 27
1.4.3 Knowledge infusion through a semantic loss function for context-aware HAR 30
1.4.4 Explainable deep learning classifiers for sensor-based HAR 32
1.5 Outline 34

2 Related work **36**

2.1 Human Activity Recognition (HAR) 36
2.1.1 Sensor-based HAR 37
2.2 Methods for sensor-based HAR 38
2.2.1 Data-driven methods 38
2.2.2 Knowledge-based methods 40
2.2.3 Neuro-Symbolic AI (NeSy) methods 41

2.3 The labeled data scarcity issue . . . 43
2.3.1 Labeled data scarcity in multi-subject HAR . . . 43
2.3.2 Labeled data scarcity in context-aware HAR . . . 45
2.3.3 Mitigating labeled data scarcity in HAR . . . 45
2.4 The lack of interpretability issue . . . 47
2.4.1 XAI taxonomy . . . 47
2.4.2 Evaluating the effectiveness of explanations . . . 48
2.4.3 XAI in activity recognition . . . 49
2.5 Research problems addressed by this thesis . . . 50

3 Neuro-symbolic HAR in multi-subject smart-home environments 53
3.1 Introduction . . . 53
3.2 The data association problem . . . 55
3.3 MICAR's architecture . . . 56
3.4 MICAR under the hood . . . 58
3.4.1 Sensing sources . . . 58
3.4.2 Semantic-data aggregation . . . 58
3.4.3 Symbolic data association . . . 61
3.4.4 Sensor-based activity recognition . . . 63
3.4.5 Prediction refinement . . . 65
3.4.6 Predictions aggregation . . . 66
3.4.7 Prediction confidence evaluation . . . 67
3.5 Experimental evaluation . . . 68
3.5.1 The MARBLE dataset . . . 68
3.5.2 Evaluation methodology . . . 73
3.5.3 Results . . . 74
3.6 Discussion . . . 81
3.6.1 Acceptability and privacy issues . . . 81
3.6.2 Personalization . . . 81
3.6.3 Need for real-world experiments . . . 82
3.7 Summary . . . 83

4 Knowledge infusion through symbolic features for context-aware HAR 84
4.1 Introduction . . . 84
4.2 Preliminaries . . . 86
4.2.1 Context-Aware Human Activity Recognition . . . 86
4.2.2 Neuro-Symbolic Context-aware HAR . . . 87
4.2.3 Formalization of existing Neuro-Symbolic approaches . . . 90
4.3 Knowledge infusion through symbolic features . . . 92
4.4 Ontological models . . . 94
4.4.1 Translating context data into ontological facts . . . 95
4.4.2 Standard ontology . . . 95
4.4.3 Probabilistic ontology . . . 98
4.5 Experimental evaluation . . . 99
4.5.1 Datasets . . . 100
4.5.2 Experimental setup . . . 102
4.5.3 Results . . . 107
4.6 Discussion . . . 112
4.6.1 Context data collection . . . 112
4.6.2 Generalizability of the approach . . . 113
4.7 Summary . . . 113

5 Knowledge infusion through a semantic loss function for context-aware HAR 115
5.1 Introduction . . . 115
5.2 Knowledge infusion through semantic loss . . . 117
5.2.1 Methodology . . . 117
5.3 Experimental evaluation . . . 121
5.3.1 Experimental setup . . . 121
5.3.2 Results . . . 122
5.4 Discussion . . . 131
5.4.1 Strengths and weaknesses of Neuro-Symbolic approaches . 131
5.4.2 Revising/updating the knowledge model . . . 132
5.4.3 Interpretability . . . 133

5.5 Summary 135

6 Explainable deep learning classifiers for sensor-based HAR 137

6.1 Introduction 137
6.2 Methodology 140
6.2.1 Deriving semantic states from sensor data 140
6.2.2 Segmentation 144
6.2.3 Information about past activities 145
6.2.4 Image generation 145
6.2.5 Deep XAI approaches 147
6.2.6 Generating semantic explanations in natural language . . . 152
6.3 Experimental evaluation 155
6.3.1 Datasets 156
6.3.2 Evaluation methodologies 158
6.3.3 Results 160
6.3.4 Impact of pre-processing hyper-parameters 170
6.4 Discussion 171
6.4.1 Over-reliance in explanations 172
6.4.2 Limitations of the *Model Prototypes* approach 172
6.5 Summary 173

7 Conclusion 176

7.1 Summary 176
7.2 Future work 180

List of Figures

1.1 Our illustrated vision of an ideal Neuro-Symbolic AI framework for sensor-based HAR 22

2.1 Pipeline typically adopted by sensor-based HAR approaches based on standard machine learning classifiers 38
2.2 Pipeline typically adopted by sensor-based HAR approaches based on deep learning classifiers 39

3.1 Overall architecture of MICAR 56
3.2 A simplified representation of a small portion of our ontology. Each node encodes an entity, while each edge encodes a relationship. . . 63
3.3 The simulated smart home during the data collection process of MARBLE 70
3.4 Evolution of the recognition rate of MICAR 74
3.5 MICAR vs a fully supervised approach 74
3.6 Confusion matrix 75
3.7 Impact of the cache on the evolution of the recognition rate . . . 76
3.8 Impact of the cache on the evolution of the percentage of active learning queries 76
3.9 The recognition rate obtained by our data association approach with respect to a naive solution and an ideal solution 77
3.10 The percentage of questions obtained by our data association approach with respect to a naive solution and an ideal solution . . . 77
3.11 The impact of our prediction refinement approach on the recognition rate 78

3.12 The impact of our prediction refinement approach on the percentage of questions 78
3.13 Confusion matrix on the number of users attributed to group activities 80

4.1 The neuro-symbolic context-aware HAR approach 89
4.2 The *context refinement* neuro-symbolic approach [1]. In this example, two activities are excluded from the probability distribution since their likelihood, according to the Symbolic Reasoner module, is 0. 90
4.3 The *symbolic features* neuro-symbolic approach 92
4.4 Excerpts of our standard ontology 96
4.5 Examples of activity definitions in our ontology 97

5.1 Our neuro-symbolic approach based on semantic loss 118
5.2 Comparison between the confusion matrices of the *baseline* and the three considered Neuro-Symbolic AI approaches trained with 10% of training data on the *ExtraSensory* dataset 128
5.3 Average feature importance for the *on transport* activity obtained using XAI methods on the **baseline** model. The brighter the color, the more important the corresponding feature was for classification. 134
5.4 Average feature importance for the *on transport* activity obtained using XAI methods on the **semantic loss** model. The brighter the color, the more important the corresponding feature was for classification. 134
5.5 Example explanation for a sample of the *walking* activity based on the x-axis measurements from the smartwatch's accelerometer. The brightness of the color indicates the level of importance of each measurement for classification. 135

6.1 The overall data flow of DeXAR 141

6.2 An example of an image generated from the MARBLE dataset [2] (more details will be presented in Section 6.3) related to the *eating* activity. The sub-matrix on the left encodes the temporal relationships of semantic states. For instance, in this temporal window of 16 seconds, the user was mostly in the dining room sitting on the dining room chair. Also, he performed some *dynamic* hand gestures, probably to eat. The sub-matrix on the right shows information about past activities. In this case, there is only one previous activity: *cooking a hot meal* 146
6.3 Grad-CAM: An example of an explanation for an input related to the *eating* activity. The most important aspects (in yellow) are the location in the dining room and the fact that the user previously performed *cooking a hot meal*. Grad-CAM also finds out that is relatively important (light green) that the subject was previously in the kitchen and that he was sitting on the chair of the dining room. Even though the resident performed hand manipulations, these are associated with low importance 149
6.4 LIME: An example of an explanation for an input related to the *eating* activity. LIME deduced that the most important feature for classification was that the resident previously performed the activity *cooking a hot meal* 150
6.5 Model Prototypes: An example of an explanation for an input related to the *eating* activity, and the m-closest prototypes learned by the DL classifier. By comparing the prototypes and the input, the most important feature in the explanation is the presence of the resident in the dining room. Also, Model Prototypes deduced that sitting on the dining room chair was also relatively important 152
6.6 Recognition rate of the low-level activities classifier (leave-one-subject-out cross-validation) 157

6.7 A small portion of our semantic model based on common-sense knowledge. Cooking devices (like the cooker, the microwave, and the oven) partially explain only the *preparing a hot meal* activity. On the other hand, kitchen repositories (that may also include the refrigerated ones), partially explain both *preparing a hot meal* and *preparing a cold meal* activities 159
6.8 A screenshot from our survey 161
6.9 MARBLE: Comparison of the recognition rates obtained by the different CNNs. *CNN-GL* is the model used for *Grad-CAM* and *LIME*, while *CNN-MP* is the one used for *Model Prototypes* with 500 learned prototypes. 163
6.10 CASAS: Comparison of the recognition rates obtained by the different CNNs. *CNN-GL* is the model used for *Grad-CAM* and *LIME*, while *CNN-MP* is the one used for *Model Prototypes* with 100 learned prototypes. 164
6.11 Overall explanation score obtained by the different XAI approaches based on the common-sense knowledge evaluation 165
6.12 MARBLE: Explanation score for each activity obtained by the different XAI approaches based on the common-sense knowledge evaluation. 167
6.13 CASAS: Explanation score for each activity obtained by the different XAI approaches based on the common-sense knowledge evaluation. 167
6.14 Distribution of the scores provided by the participants for each method 169
6.15 Comparison between common-sense knowledge evaluation and user-based evaluation 169
6.16 Impact of the segmentation window dimension (in seconds) on the recognition rates 170
6.17 Impact of K (number of past activities) and t (consecutive reliable predictions) on the recognition rates 171
6.18 Impact of c (confidence threshold for past activities) on the recognition rates 171

List of Tables

3.1 A scenario involving two subjects 72
3.2 Statistics on labeled activities 72
3.3 Statistics on scripted scenarios 73

4.1 Number of samples for each activity class in *DOMINO* 103
4.2 Number of samples for each activity class in *ExtraSensory* 104
4.3 DOMINO: Results in terms of macro F1 score and 95% confidence interval 108
4.4 ExtraSensory: Results in terms of macro F1 score and 95% confidence interval 109
4.5 Average results with 5 different runs in terms of macro F1 score, considering 10% of training data and different percentages of dirty samples in the test set 111
4.6 Average results with 5 different runs in terms of macro F1 score, considering a data scarcity scenario simulated by using 10% of training data and the probabilistic version of each method 111

5.1 Comparison between the Semantic Loss types on the different datasets 123
5.2 DOMINO: Results in terms of macro F1 score and 95% confidence interval 125
5.3 ExtraSensory: Results in terms of macro F1 score and 95% confidence interval 126
5.4 Average results with 5 different runs in terms of macro f1 score, considering 10% of training data and different percentages of dirty samples in the test set 130

5.5 Average results with 5 different runs in terms of macro F1 score, considering a data scarcity scenario simulated by using 10% of training data and the probabilistic version of each method 130
5.6 Comparison of pros and cons of NeSy methods 131

6.1 Hyperparameters in DeXAR 162

Chapter 1

Introduction

1.1 Motivation

In the last few years, technological progress has led to the widespread availability of cost-effective and sensor-equipped devices with computing and communication capabilities. This advancement coupled with the efforts in the *ubiquitous sensing* research area, whose purpose is to extract knowledge from the data collected by pervasive sensors, led to the development of pervasive and context-aware applications [3]. Pervasive systems aim to ubiquitously assist users in fulfilling their tasks, exploiting data provided by the sensors built into the smart devices that are embedded in our living spaces. Among pervasive systems, there are context-aware applications, i.e., solutions that adapt their behavior based on the users' surrounding context (e.g., the time of the day, local weather conditions, or the activities the users are performing).

Context-aware applications based on Human Activity Recognition (HAR) frameworks aspire to detect the activities performed by the users and use such information to enhance the services they provide to them. Hence, HAR solutions enable applications in several domains, including surveillance, security, well-being, and healthcare [4]. For instance, HAR can be exploited to monitor the physical and cognitive health of home-based patients [5], thus early detecting and preventing the emergence of medical conditions.

1.2 Problem definition

Overall, researchers in the HAR domain mainly focused on the detection of activities that fall into two main categories: *low-level (physical) activities* (e.g., standing, sitting on a bus, walking, or running), and *high-level activities* like Activities of Daily Living (ADLs) (e.g., cooking, taking the medicines, or watering the plants) or the activities that can be performed in a working environment (e.g., in a meeting). According to the adopted sensing infrastructure, different methods have been proposed in the literature to detect both activity categories. Most of the existing works proposed video-based or sensor-based solutions [6]. Video-based methods detect users' activities by processing videos that are typically captured by cameras. However, despite the promising results reached with video-based approaches, the need to continuously monitor users through cameras raises significant privacy concerns. While these issues can be mitigated thanks to privacy-preserving hardware/software techniques, cameras are generally perceived as too intrusive by users. Consequently, different research groups preferred to focus their efforts on sensor-based HAR, a less invasive alternative [7].

In sensor-based HAR, the adopted sensing technology affects the category of activities that can be recognized. Wearable devices like smartphones and smartwatches that are equipped with inertial sensors help in monitoring the users' body movements at a fine granularity. Hence, they are typically used to classify low-level activities [8]. Moreover, personal wearable devices can also provide users' contextual information: for instance, sensor data (e.g., the GPS coordinates collected by the smartphone) and external services like Google's Places API can be combined to derive the users' closest semantic places (e.g., their workplace, a gym) [1]. Context data are hence useful to expand the set of recognizable activities by discriminating those with similar motion patterns (e.g., sitting and sitting on a train).

On the other hand, smart environments (e.g., smart homes/offices) equipped with environmental sensors like smart plugs and motion sensors allow HAR applications to monitor the interactions of the users with their surroundings [9]. This information can be exploited to recognize high-level activities like ADLs. HAR

researchers mainly explored smart environments occupied by a single user, even if there are also many real-world cases where multiple subjects are present in the same space (e.g., co-workers in a smart office, or elderly subjects that cohabit with their caregivers in a smart home). Multi-subject settings are significantly more challenging than single-subject ones. One of the major open problems is *data association*, i.e., correctly associating environmental sensor events (like the opening of the fridge) with the subject(s) that actually generated them. Solving data association would ease the recognition of the users' activities that can be performed either individually (e.g., Alice is reading, while Bob is watering the plants) or collaboratively (e.g., Alice and Bob are cooking together).

Most sensor-based HAR approaches proposed in the literature rely on supervised Deep Learning (DL) models since they reach high recognition rates, overcoming some limitations of more conventional Machine Learning (ML) solutions. Indeed, ML methods for sensor-based HAR depend on heuristic and handcrafted feature extraction procedures that rely on human domain knowledge. Therefore, the extracted features usually include only statistical information about the collected data. On the other hand, DL models can automatically learn features that also encode high-level representations of data, thus making DL more suitable for complex HAR tasks [7]. Despite their success, the deployment of DL models in real-world scenarios is limited by research issues that are still open. For instance, during their learning process, such models require large amounts of labeled training data that are challenging to annotate (i.e., the *data scarcity* problem). Indeed, data annotation is an error-prone, expensive, tedious, and time-consuming procedure [10]. Moreover, the decision-making process of DL models is inherently opaque. This does not allow humans to understand the rationale behind each model's prediction [11]. Explainable Artificial Intelligence (XAI) is hence becoming a popular strategy to make DL models more transparent. However, it is challenging to generate meaningful explanations for predictions based on sensor data as well as to evaluate their effectiveness to the target users.

In the sensor-based HAR literature, purely knowledge-based approaches have been considered to tackle both the lack of transparency and the labeled data scarcity issues [8]. Such methods rely on reasoning (e.g., through logic rules) over

a symbolic representation of the HAR domain that is modeled based on human common sense and domain knowledge. For instance, *washing the dishes* can be defined as an activity that is typically performed while standing in the kitchen close to the sink, after eating. In this way, the most likely activities can be derived by matching symbolic rules with sensor events and users' contextual information. Symbolic approaches present two main advantages: (i) they are based on human-readable formalisms that make them transparent and interpretable, and (ii) they do not require any labeled data sample. However, these techniques are too rigid and not scalable since it is unlikely to take into account logic constraints that cover all the possible ways in which activities can be performed. Moreover, they are not suitable for sensors that generate continuous values like accelerometers. Indeed, such raw sensor measurements cannot be mapped to a clear semantic, thus making it impossible to include them in any symbolic rule.

Recently, Neuro-Symbolic AI (NeSy) solutions have emerged in the general ML community to combine the strengths of data-driven and knowledge-based methods [12]. The overall goal is to enhance DL models through domain knowledge to achieve several potential benefits. To begin, NeSy methods may significantly improve the recognition rates by driving classifiers with domain constraints. This may be especially true when only limited amounts of labeled data are available: in these cases, those constraints cannot be learned directly from data. For instance, according to common-sense knowledge, the activity *offline shopping* is typically performed in specific semantic locations (e.g., shops, commercial areas). This intuitive association can be represented using a symbolic formalism and infused into the DL model, thus reducing the amount of labeled data required to learn it. Similarly, domain knowledge may improve the recognition of those cases out of the training set distribution samples. Moreover, influencing the decisions of DL models through human knowledge can make them intrinsically more interpretable and transparent [13].

1.3 An ideal neuro-symbolic framework for HAR

In this thesis, we will focus on novel NeSy methods for sensor-based HAR. Their goal is to detect low- or high-level activities by relying on the data collected

through the users' personal wearable devices and/or environmental sensors installed within their living spaces. Figure 1.1 presents our vision of an ideal NeSy framework for sensor-based HAR, where DL classifiers and symbolic reasoning

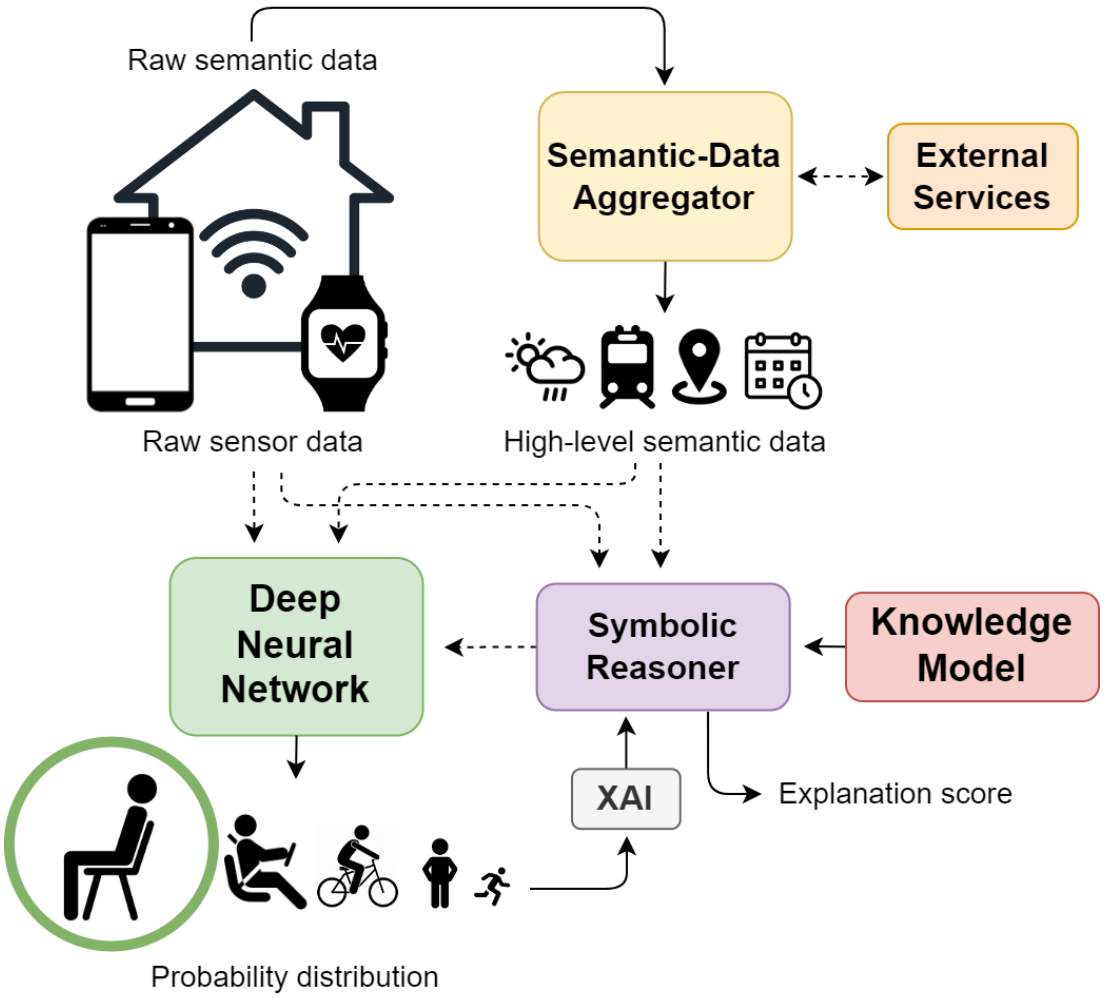


Figure 1.1: Our illustrated vision of an ideal Neuro-Symbolic AI framework for sensor-based HAR

are coupled (i) to detect users' activities by relying on relatively lower amounts of labeled data compared to purely data-driven approaches, (ii) to make the solution inherently interpretable by exploiting knowledge-based reasoning, and (iii) to quantitatively evaluate the degree of consistency of the explanations (obtained through XAI methods) with HAR domain knowledge. In this ideal NeSy

framework, data collected from wearable devices and/or environmental sensors are segmented into fixed-length windows. The data included in each window can be divided into two possibly overlapping subsets: raw sensor data and raw semantic data. Raw sensor data (e.g., accelerometer measurements) are the ones that are appropriate to be directly processed by a DEEP NEURAL NETWORK to automatically extract meaningful features. On the other hand, raw semantic data are sensor measurements that a SEMANTIC-DATA AGGREGATOR can use to derive high-level semantic information about the user, by relying on simple rules or EXTERNAL SERVICES like context-aware middlewares [14] and web services. Intuitively, high-level semantic data should enable knowledge-based reasoning. For instance, raw GPS coordinates collected by a smartphone can be used by the SEMANTIC-DATA AGGREGATOR to derive the semantic location of the user through the interaction with public web services (e.g., Google's Places API); hence, knowledge-based reasoning could be used to find the activities that are typically being performed in the current semantic location of the user (e.g., brushing teeth is typically performed in spaces that are familiar to the users, like their home or workplace).

According to the application of interest, the NeSy framework can use raw sensor data and high-level semantic data in different ways. For instance, some applications may require high-level semantic data only to perform knowledge-based reasoning through the symbolic reasoner. In other cases, such data can also be provided as input to the DEEP NEURAL NETWORK.

Overall, the DEEP NEURAL NETWORK finds correlations between input data and activities in a data-driven way. On the other hand, the SYMBOLIC REASONER performs knowledge-based reasoning to match its input data with the domain constraints encoded into a KNOWLEDGE MODEL. An example of domain constraint is that, in a multi-subject smart home, only the residents who are currently in the kitchen can turn the stove on. As we will see in this thesis, the SYMBOLIC REASONER can be designed to solve various HAR tasks, and its output can be used in different ways to reduce the amounts of labeled data required by the DEEP NEURAL NETWORK to reliably recognize users' activities. Moreover, compared to approaches only based on deep learning, our ideal NeSy framework is inherently more interpretable since its decision-making process is also driven

by knowledge-based reasoning.

Finally, the SYMBOLIC REASONER can also be exploited to evaluate through a score the explanations generated by XAI methods applied to the predictions made by the DEEP NEURAL NETWORK. More specifically, this explanation score quantitatively measures the degree of consistency of such explanations with HAR domain knowledge. This would give an assessment of the framework's interpretability level.

1.4 Research contributions

In this section, we investigate some of the main issues related to state-of-the-art approaches for sensor-based HAR that limit their deployment in real-world scenarios. Then, we introduce every research contribution of the thesis with the goal of getting closer to our vision for an ideal NeSy framework. It is important to note that these contributions have been achieved in collaboration with my research group, i.e., the EveryWare Lab[1], at the University of Milan (Italy).

1.4.1 Neuro-symbolic HAR in multi-subject smart-home environments

One of the main issues of state-of-the-art sensor-based HAR approaches is the labeled data scarcity problem. Indeed, collecting and annotating sufficient amounts of training data to build scalable DL activity classifiers that generalize across different types of users and smart environments is a real challenge. For instance, users involved in data collection campaigns can directly annotate their own data while performing activities. However, this approach is particularly error-prone since users can forget to annotate relevant activities or the exact time in which they were performed [15]. This negatively impacts the quality and reliability of the annotated data. Alternatively, external observers can annotate activity data by monitoring the subjects involved during data acquisition. Unfortunately, this solution can be expensive and time-consuming, even if performed through

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

cameras and semi-automatic video annotation tools. Moreover, constantly monitoring users is privacy-invasive, especially considering private smart environments like smart homes.

The labeled data scarcity issue can be further emphasized in applications that recognize users' activities through environmental sensors installed in smart environments occupied by multiple subjects. In these scenarios, environmental sensors cannot automatically identify the user(s) that trigger them. For instance, a pressure mat sensor on a chair cannot reveal the user sitting on it. The process of mapping environmental events to the correct user is called data association, and it is essential to reliably infer the activities performed by each user in the smart environment [16]. Most of the existing literature tackled data association in a data-driven fashion, thus aggravating the labeled data scarcity problem. Many solutions involve ML or DL methods that require training sets containing samples of all the possible combinations of activities that users can potentially perform together or individually [17]. Sometimes, data association is instead considered as a separate learning problem before activity classification. In particular, some research groups performed a weaker form of data association (named *resident separation*), by investigating unsupervised solutions that identify pairs of environmental sensors' events triggered by the same resident, without identifying her [18]. On the other hand, other works require additional labeled data about users' habits to train a supervised classifier that associates each sensor event with a specific set of identified users [19].

To mitigate the data scarcity problem, in Chapter 3 we propose a novel NeSy approach that relies on symbolic reasoning to perform data association by combining users' contextual information (e.g., their posture and location in the environment) with triggered sensor events [20, 21]. In this way, annotated data samples can be separated into a personalized stream of sensor events for each user, without requiring additional labeled data. Such streams are then used to train an activity classifier. Hence, symbolic reasoning is involved also after deployment to produce these personalized streams of sensor events that the classifier receives to predict the activities performed by each user.

In particular, to further mitigate data scarcity, we developed and experimen-

tally evaluated a semi-supervised activity classifier. This classifier is an incremental model initialized with a limited amount of labeled data. Hence, a cache-based active learning strategy is adopted over time to collect novel annotated data samples that are exploited to continuously improve such a model. In the proposed framework, symbolic reasoning is also used to perform *context refinement* [1], i.e., to refine the predictions of the classifier by discarding from the probability distributions it generates those activities that are not consistent with the users' contextual information (e.g, only the residents that are currently in the kitchen can cook).

Our results on MARBLE [2], a dataset we collected and published where up to 4 subjects perform activities at the same time in the same smart environment, show how the proposed framework reliably recognizes individual and collaborative activities, without requiring any additional labeled data to perform data association. In particular, the semi-supervised classifier reaches similar recognition rates compared to a fully-supervised model, while requiring significantly lower labeled data and triggering a limited number of active learning queries. Moreover, context refinement based on symbolic reasoning further improves the classifier recognition rates and reduces the active learning queries required by the semi-supervised approach.

Chapter 3 is based on the following publications:

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"MICAR: Multi-Inhabitant Context-Aware Activity Recognition in Home Environments"*. Distributed and Parallel Database, Springer, 2022. (DOI: 10.1007/s10619-022-07403-z)
- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"The MARBLE Dataset: Multi-Inhabitant Activities of Daily Living Combining Wearable and Environmental Sensors Data"*. In International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (MobiQuitous). Cham: Springer International Publishing, 2021.
- 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.

Scientific contributions:

- Introduction of a novel neuro-symbolic AI framework for multi-subject activity recognition.
- Use of symbolic reasoning to perform data association without additional labeled data.
- Presentation of a novel cache-based active learning strategy to further mitigate labeled data scarcity.
- Experiments on a public multi-subject dataset show (i) how the proposed framework is comparable to a fully supervised solution in terms of recognition rates, and (ii) how the accuracy of data association based on symbolic reasoning is close to the one of an ideal approach based on ground truth.

Personal tasks:

- Collaboration in concept and methodology design.
- Method implementation.
- Collaboration in the design of the evaluation methods.
- Experiments execution.
- Collaboration in results analysis and interpretation.

1.4.2 Knowledge infusion through symbolic features for context-aware HAR

Labeled data scarcity also affects other application domains, like the context-aware recognition of low-level (physical) activities through mobile/wearable de-

vices. In this scenario, researchers introduced the use of contextual information about the user's surroundings (e.g., semantic location, speed, and weather conditions) [22] that has the potential to better discriminate activities with similar motion body movements (e.g., standing and getting an elevator). The disadvantage of this approach is that it is not realistic to acquire a comprehensive training set that includes every possible context condition in which activities can be performed by different types of users.

Existing NeSy methods in the literature already mitigated this problem [23, 1]. However, like the context refinement approach mentioned in Section 1.4.1, they only consider domain knowledge to discard from the output of the activity classifier those activities that are not consistent with the user's surrounding context. Approaches of this kind can make wrong decisions when the knowledge model does not cover all the main context scenarios in which activities can be carried out by users. For instance, if a user runs within a mall and the knowledge model does not take into account such a scenario, the *running* activity would be discarded by these approaches. The same problem arises in the presence of temporary noisy contextual information: for instance, GPS readings from the user's smartphone could be momentarily noisy, thus leading to incorrect contextual information about the user; hence, existing NeSy methods could improperly discard the wrong activities.

This problem can be mitigated through Knowledge Infusion, i.e., an emerging NeSy approach that infuses domain knowledge directly into the DL classifier during training. In this way, the model internally learns and correlates domain constraints with user activities and the other input data, while handling data uncertainty thanks to its data-driven learning process. In Chapter 4, we propose a novel knowledge infusion method for context-aware HAR. The features automatically extracted by the DL classifier from raw sensor data and high-level semantic data are combined with the ones inferred through symbolic reasoning. Such symbolic features encode domain knowledge about the activities that are consistent with the user's surrounding context and they are infused within the DL model, before the classification layer. We implemented two versions of this NeSy approach. In the first case, symbolic reasoning relies on a standard ontol-

ogy encoding hard constraints between context information and activities. For instance, the activity *running* implies that the current user's speed is positive. In the second case, we consider a probabilistic ontology composed of both hard and soft constraints (i.e., rules associated with a weight). For instance, the soft constraint *running can be performed indoors* has a lower weight than the soft constraint *running can be performed outdoors*.

Our results on DOMINO [24], a dataset for context-aware HAR we recently published, and on another real-world context-aware HAR dataset show how the use of symbolic features mitigates data scarcity while being more robust than context refinement in the presence of noisy context data. Moreover, we show how the improvements led by probabilistic ontologies do not justify the significant effort required to build them.

Chapter 4 is based on the following publications:

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"Semantic Loss: a new Neuro-Symbolic approach for Context-Aware Human Activity Recognition"*. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2023, to appear.

- Luca Arrotta, Gabriele Civitarese, Riccardo Presotto, Claudio Bettini, *"DOMINO: A Dataset for Context-Aware Human Activity Recognition using Mobile Devices"*. In 2023 24th IEEE International Conference on Mobile Data Management (MDM) Workshops. IEEE, 2023.

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"Probabilistic Knowledge Infusion through Symbolic Features for Context-Aware Activity Recognition"*. Pervasive and Mobile Computing, Elsevier, 2023.
  (DOI: 10.1016/j.pmcj.2023.101780)

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"Knowledge Infusion for Context-Aware Sensor-Based Human Activity Recognition"*. In 2022 IEEE International Conference on Smart Computing (SmartComp), 2022.

Scientific contributions:

- Introduction of a novel knowledge infusion method for context-aware HAR to improve the latent space representation of sensor and context data with symbolic features based on domain and common-sense knowledge.
- Experiments on two public datasets show how the proposed framework (i) outperforms a purely data-driven classifier and (ii) is more robust in the presence of noisy context data compared to state-of-the-art neuro-symbolic AI solutions for HAR.

Personal tasks:

- Concept and methodology design.
- Method implementation.
- Collaboration in the design of the evaluation methods.
- Experiments execution.
- Collaboration in results analysis and interpretation.

1.4.3 Knowledge infusion through a semantic loss function for context-aware HAR

In Chapter 4, we proposed a novel Knowledge Infusion method for context-aware HAR, comparing it with state-of-the-art NeSy solutions. However, all the previously introduced methods require symbolic reasoners during classification. In real-world deployments, where the DL model can be deployed on resource-constrained devices (e.g., mobile/wearable devices), the adoption of symbolic reasoning during classification is not desirable since it is computationally demanding [25].

For this reason, in Chapter 5, we propose a novel Knowledge Infusion approach

based on a semantic loss function that infuses knowledge constraints in the HAR model only during training, avoiding symbolic reasoning after deployment. In particular, we implemented a custom loss function for the DL model combining a standard classification loss with a novel semantic loss function. The semantic loss component uses symbolic reasoning to drive the DL model in classifying activities considering domain knowledge constraints. After the training phase, the classifier internally encodes such constraints, that are exploited to classify activities at run-time without requiring symbolic reasoning.

Our results on scripted and in-the-wild datasets show the impact of different semantic loss functions (that rely on a standard or a probabilistic ontology) in outperforming a purely data-driven model. We also compare our solution with existing NeSy methods (including the one proposed in Chapter 4) and analyze each approach's strengths and weaknesses. Our method based on a semantic loss remains the only NeSy solution that can be deployed without the need for symbolic reasoning modules, reaching recognition rates close (and better in some cases) to existing approaches. Moreover, our results demonstrate how our semantic loss is significantly more robust than the other NeSy approaches in the presence of noisy data. Finally, we also briefly inspect interpretability aspects, qualitatively showing how our semantic loss method makes decisions following the domain constraints encoded into the infused knowledge. This result is a first step that indicates how NeSy methods can lead to more interpretable DL models.

Chapter 5 is based on the following publications:

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"Semantic Loss: a new Neuro-Symbolic approach for Context-Aware Human Activity Recognition"*. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2023, to appear.

Scientific contributions:

- Introduction of a novel knowledge infusion method for context-aware HAR based on a semantic loss function that does not require symbolic reasoning after the HAR system deployment.
- Experiments on two public datasets show how our semantic loss method (i) outperforms a purely data-driven classifier and (ii) is significantly more robust than other neuro-symbolic approaches in the presence of noisy data.

Personal tasks:

- Collaboration in the problem formulation.
- Concept and methodology design.
- Method implementation.
- Collaboration in the design of the evaluation methods.
- Experiments execution.
- Collaboration in results analysis and interpretation.

1.4.4 Explainable deep learning classifiers for sensor-based HAR

In Chapter 5, we made a first step towards the analysis of possible interpretability benefits provided by NeSy methods. However, in the current sensor-based HAR literature, no quantitative metric has been introduced to measure the interpretability level of DL models. This problem is due to the fact that it is challenging to apply eXplainable AI (XAI) methods to raw sensor data since most XAI techniques in the literature are focused on computer vision tasks. Indeed, the few works that explored XAI methods for HAR only considered interpretable machine learning models.

In Chapter 6, we propose a novel methodology to transform sensor data to take advantage of XAI methods designed for computer vision tasks. We then apply different XAI approaches for deep learning and, from the resulting heat maps, we generate explanations in natural language. In order to identify the most effective XAI method, we design a metric (i.e., the Explanation Score) that measures the coherence of such explanations with human knowledge about the HAR domain.

Our results show how the evaluations performed through the Explanation Score are aligned and consistent with the ones obtained through a user-based evaluation (i.e., a survey). Unfortunately, due to time constraints, we have currently used the Explanation Score only to evaluate purely data-driven approaches. Nonetheless, we believe that the promising results presented in Chapter 6 indicate that this metric could be considered in the future to quantify the interpretability benefits provided by NeSy methods for sensor-based HAR.

Chapter 6 is based on the following publications:

- Luca Arrotta, Gabriele Civitarese, Claudio Bettini, *"DeXAR: Deep Explainable Sensor-Based Activity Recognition in Smart-Home Environments"*. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2022. (DOI: 10.1145/3517224)

Scientific contributions:

- Introduction of a novel XAI framework for sensor-based HAR that relies on deep learning models.
- Design of a metric that measures the coherence of XAI explanations with HAR domain knowledge.
- Experiments on a public dataset show how the results obtained through the proposed metric are consistent with the ones of a user-based evaluation.

Personal tasks:

- Collaboration in the problem formulation.
- Collaboration in the methodology design.
- Method implementation.
- Collaboration in the design of the Explanation Score
- Collaboration in the design of the evaluation methods.
- Experiments execution.
- Collaboration in results analysis and interpretation.

1.5 Outline

The rest of the thesis is structured as follows. Chapter 2 provides a wide overview of the current literature on sensor-based HAR, introducing the specific challenges tackled by this thesis. Chapter 3 presents a novel NeSy HAR framework that relies on symbolic reasoning to perform data association without labeled data in multi-subject smart homes. In chapters 4 and 5, we present novel NeSy methods based on Knowledge Infusion for context-aware HAR. In particular, Chapter 4 presents a Knowledge Infusion approach that relies on symbolic reasoning to infer

additional knowledge-based features to be infused into the DL activity classifier. Such features guide the model to learn with fewer training samples the correlations between input data and users' activities. On the other hand, Chapter 5 presents another Knowledge Infusion method based on a semantic loss function to infuse domain knowledge into the DL classifier only during training, thus avoiding any computationally demanding symbolic reasoning step after deployment. Chapter 6 introduces a novel XAI framework for sensor-based HAR in smart homes based on deep learning. This chapter also presents the Explanation Score, a quantitative metric that measures how much XAI explanations are aligned with domain knowledge. Finally, Chapter 7 summarizes our contributions, outlines future research direction, and concludes this thesis.

Chapter 2

Related work

2.1 Human Activity Recognition (HAR)

In the last decade, Human Activity Recognition (HAR) has become a task of high interest since it enables pervasive and context-aware applications that adapt their services based on information about the users, such as their habits, behavior, and health status. [3]. HAR systems commonly monitor users to derive the activities they perform thanks to the data collected by a variety of sensors [26]. In particular, the existing literature mainly focused on video-based and sensor-based HAR [27]. Video-based methods analyze videos obtained from optical sensors like cameras [28]. This information-rich data type leads to very accurate HAR solutions. However, the deployment of these methods is limited in many environments (e.g., private habitations) since cameras are generally perceived as too intrusive by the monitored users [8]. On the other hand, in sensor-based HAR, data are usually collected through inertial sensors like accelerometers embedded into wearable devices or through environmental sensors (e.g., magnetic sensors, smart plugs) installed in the users' living spaces [7]. In the last few years, the proliferation of cheap, ubiquitous, and non-intrusive IoT devices led many research groups to concentrate on sensor-based HAR [6]. Accordingly, in this thesis, we will focus on these kinds of methods.

2.1.1 Sensor-based HAR

Overall, the HAR literature focused on the recognition of two types of activities: low-level (e.g., walking, taking the stairs) and high-level activities (e.g., having a meeting, eating). In sensor-based HAR, the adopted sensing infrastructure significantly affects the types of activities that can be recognized [8].

Low-level activities are mostly characterized by the user's physical movements. Therefore, these kinds of activities are typically recognized through wearable devices like smartphones or fitness bands. Indeed, these devices are equipped with inertial sensors (e.g., accelerometers, gyroscopes) that help in monitoring the users' body movements at a fine granularity. Moreover, personal wearable devices like smartphones can collect contextual information about the users' surroundings. For instance, mobile apps installed on the user's device can interact with public web services to collect information about local weather conditions. These kinds of data are helpful to discriminate activities with similar motion patterns typically performed in different context scenarios [1].

Differently, high-level activities (e.g., cooking) also involve interactions of the users with their surroundings. Consequently, these activities typically require the installation in the users' living spaces of environmental sensors able to capture such interactions. For instance, a pressure mat sensor can reveal that the user is currently sitting at the dining table, thus easing the recognition of the eating activity. Overall, HAR researchers mainly investigated smart environments occupied by a single user [8]. However, many real-world applications involve multiple users performing activities in the same shared space. For instance, elderly subjects can live with their partners in a smart home. One of the major problems in multi-subject smart environments is that environmental sensors cannot automatically identify the user that triggered them. For instance, a magnetic sensor attached to the fridge cannot reveal the users that opened it. The process of mapping environmental events to the correct user is called *data association*, and it is essential to reliably infer the activities performed by each user [16].

2.2 Methods for sensor-based HAR

In the following, we describe the categories of HAR methods investigated and proposed in the literature.

2.2.1 Data-driven methods

Regardless of the adopted sensor technology, sensor-based HAR has been mainly addressed with supervised data-driven methods. The goal of these approaches is to build a Machine Learning (ML) model able to recognize the users' activities based on the available sensor data.

Traditional machine learning methods

Figure 2.1 presents the typical pipeline of HAR frameworks that rely on traditional ML classifiers (e.g., random forests, support vector machines). Data

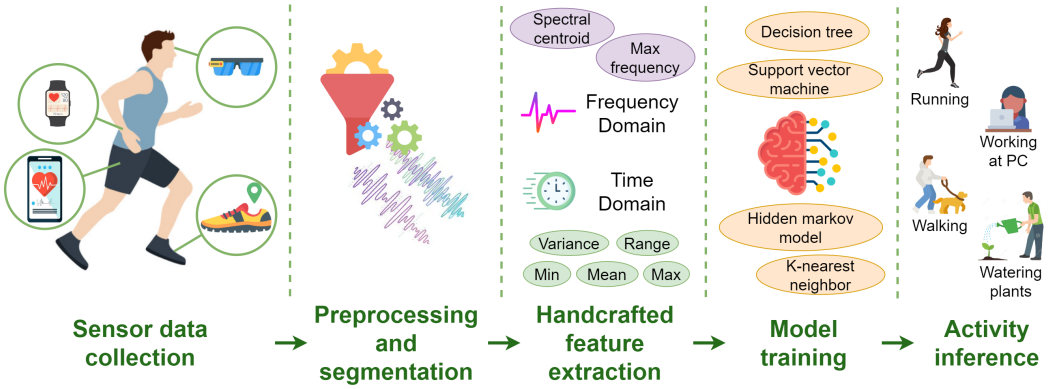


Figure 2.1: Pipeline typically adopted by sensor-based HAR approaches based on standard machine learning classifiers

streams collected from the available sensors are pre-processed (e.g., to remove noisy measurements) and partitioned into segmentation windows of a fixed size. Then, from each segmentation window, a set of features (e.g., mean, variance) is manually extracted from raw data based on heuristic and human domain knowledge. These feature vectors are finally used to train an ML model that after deployment is able to use the same features to recognize the users' activities. The most common traditional ML models proposed for HAR (considering the

recognition of both low- and high-level activities) are Decision Trees [29, 30], Support Vector Machines [31, 32], K-Nearest Neighbors [33, 34], and Hidden Markov Models [35, 36].

The main drawback of pipelines based on traditional ML models is their handcrafted feature extraction process. Indeed, human expertise can only enable the extraction of shallow features that usually include only statistical information about the collected data [7]. For instance, inertial data are typically condensed to time- and frequency-domain features like the mean and the energy of the signal, respectively [37]. On the other hand, considering environmental sensors, feature vectors typically include information like the count of the different sensor events that occurred during each segmentation window [38].

Deep learning methods

In the last years, Deep Learning (DL) models have become the leading solution for sensor-based HAR since they overcome the limitations of traditional ML classifiers. Indeed, beyond reaching high recognition rates, as depicted in Figure 2.2, DL models have the ability to automatically extract during training meaningful

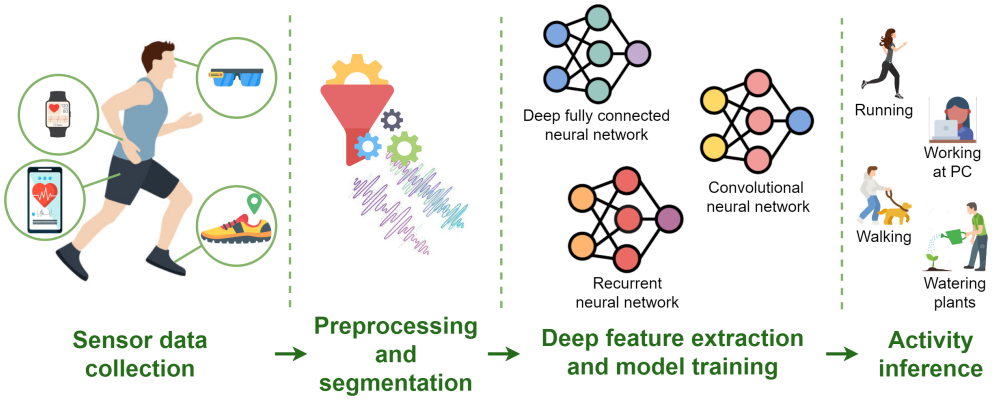


Figure 2.2: Pipeline typically adopted by sensor-based HAR approaches based on deep learning classifiers

features that are suitable to solve the HAR task. Different types of DL models have been proposed in the literature [7]. The most common are Deep fully connected Neural Networks (DNN) [39], Convolutional Neural Networks (CNN) [40, 41], Recurrent Neural Networks (RNN) [42], as well as hybrid models [43].

Despite their undeniable success, the deployment of supervised DL models is often limited by labeled data scarcity and lack of transparency, two issues that will be discussed in detail in sections 2.3 and 2.4, respectively. Indeed, supervised DL classifiers heavily rely on huge amounts of labeled samples during training. However, data annotation is a real challenge, especially in the HAR domain [10]. Moreover, the decision-making process of DL models is inherently opaque, thus not allowing humans to understand the rationale behind each model's prediction [11].

2.2.2 Knowledge-based methods

In the sensor-based HAR literature, purely knowledge-based (or symbolic) methods have been proposed to tackle the above-mentioned issues of DL models [8]. Symbolic approaches rely on formal models built by domain experts and encoding relationships between sensor events, users' contextual information, and activities [23, 44]. For instance, *brushing teeth* can be symbolically represented as an activity that is typically performed while being close to a sink in a semantic place familiar to the users (e.g., their home, their workplace). Hence, knowledge-based methods derive the most likely users' activities by reasoning (e.g., through logic rules) over the domain constraints encoded into the formal model.

Different formalisms have been proposed in the literature to encode HAR domain knowledge [45]. In particular, ontologies are the most common solution due to their expressive power and automatic reasoning capabilities [46, 47, 44]. Indeed, different research groups already built ontologies for pervasive computing [48] and activity recognition [49, 50].

Overall, symbolic approaches have two main advantages: (i) they are transparent and interpretable since based on human-readable formalisms, and (ii) they do not require labeled data for training purposes. However, these methods are too rigid and not scalable since it is not feasible to build logic constraints that cover all the possible ways and contexts in which activities can be performed. Additionally, symbolic reasoning is not suitable for those sensors that generate continuous values like accelerometers. Indeed, raw sensor measurements of this kind cannot be mapped to a clear semantic, thus making it impossible to include

them in any symbolic rule. For this reason, purely knowledge-based methods have been mainly proposed for the recognition of high-level activities in smart homes [51, 44].

2.2.3 Neuro-Symbolic AI (NeSy) methods

The goal of Neuro-Symbolic AI (NeSy) approaches is to integrate neural and symbolic AI architectures to combine their abilities to perform data-driven learning and knowledge-based reasoning [12]. This combination improves the capability of the deep learning classifier to learn from smaller amounts of training data, to better generalize on unseen data, and to increase its interpretability [52]. A promising NeSy approach is the Knowledge Infusion paradigm that aims at incorporating external knowledge (e.g., obtained from a knowledge graph) within a DL model [53, 54].

Knowledge Infusion in the general ML community

In the general ML community, in the last few years, Knowledge Infusion was mainly explored in the Computer Vision (CV) [55, 56] and the Natural Language Processing (NLP) [57, 58] domains. For instance, considering hierarchical multi-label image classification tasks, researchers explored custom loss functions to infuse into DL models information about semantic connections between classes and their hierarchy, with the objective of making misclassification less severe [59, 60, 61]. Thanks to this approach, an image labeled with the class *boy* is more likely to be misclassified with the class *man* rather than with unrelated classes like *bicycle*. Considering the NLP domain, the teacher-student learning paradigm has been explored in [62] so that a student DL model mimics the outputs of a teacher model trained with a loss function that takes into account logical rules. For instance, in sentiment analysis, a logical constraint may consider the conjunction word "but" to ensure that the predicted sentiment for the entire sentence aligns with the sentiment of the clause that follows "but".

NeSy for sensor-based HAR

Unfortunately, only a few NeSy methods exist for sensor-based HAR. In particular, the effectiveness of Knowledge Infusion for HAR is still an open research problem since, in most of the existing methods, external knowledge is only considered before [63] or after [64, 65, 23] the training process, and it is not infused into the DL model. Considering HAR in smart-home environments, domain knowledge can be used to derive an initial activity model that is subsequently adapted to the user's habits through data-driven strategies [63]. In [66], unsupervised methods are used to extract frequent patterns from unlabeled data. These patterns are then associated with the corresponding activities through domain knowledge. On the other hand, considering HAR with mobile/wearable devices, in [1], the probability distribution over the possible activities derived by a data-driven classifier is refined by common-sense knowledge constraints to exclude unlikely activities. The main drawback of the above-mentioned HAR methods is that, without Knowledge Infusion, the DL model cannot intrinsically learn domain constraints, thus limiting (i) its ability to handle data uncertainties (e.g., the model's decisions could be rigidly refined through an incomplete knowledge) and (ii) the interpretability benefits enabled when knowledge is infused. For instance, consider *context refinement* [1], i.e., a method that relies on domain knowledge to discard from the output of the activity classifier those activities that are not consistent with the user's surrounding context. This method could make wrong decisions when the knowledge model does not cover all the main context scenarios in which activities can be carried out by users. Additionally, the knowledge model cannot be used to interpret the DL classifier predictions since knowledge is not infused into it.

Neuroplex [67] is the only existing Knowledge Infusion method for sensor-based HAR. Specifically, symbolic knowledge (i.e., finite state machines and logical rules) is infused into a neural network responsible for detecting complex nursing events (e.g., *patient cleaning*). Indeed, these events can be identified by reasoning on spatially- and temporally-dependent low-level events derived from inertial sensors data using data-driven models. For instance, the complex event *patient cleaning* can be derived when the sequence of detected low-level events

is composed of *patient oral care* followed by *diaper exchange*. In this thesis, we will explore NeSy methods based on the Knowledge Infusion paradigm for HAR applications that are not covered by Neuroplex, i.e., multi-subject HAR in smart environments and context-aware HAR with mobile/wearable devices. Indeed, in multi-subject HAR, it is not straightforward to apply the idea behind Neuroplex since the considered high-level activities (e.g., cooking) can be performed by triggering several different combinations of sensor events. On the other hand, context-aware HAR aims to directly recognize the low-level physical activities (e.g., sitting) performed by the users.

2.3 The labeled data scarcity issue

As we previously mentioned, labeled data scarcity is one of the main problems that limit the deployment of DL models for HAR in real-world applications. Indeed, collecting and annotating sufficient amounts of data to train scalable supervised DL classifiers is a real challenge. Sensor data can be labeled through self-annotations or by continuously monitoring users through cameras or external observers. However, self-annotation is particularly error-prone since users can forget to label relevant activities or the exact time in which they were performed [15], thus discouraging the use of such data to train data-driven models. On the other hand, constantly monitoring users for accurate data annotation is privacy-invasive, especially in private environments (e.g., users' habitations). In this thesis, we will focus on specific HAR applications that further emphasize the labeled data scarcity problem: multi-subject HAR and context-aware HAR.

2.3.1 Labeled data scarcity in multi-subject HAR

In multi-subject smart environments, labeled data scarcity is accentuated since most of the existing methods rely on data-driven models not only to recognize users' activities but also to perform data association implicitly or in a supervised fashion. In this application scenario, purely knowledge-based approaches have been proposed to solve a weaker form of data association, called *subject separation*, where the goal is to determine whether two consecutive sensor events

were fired by the same subject or by different subjects without identifying them [68, 69]. Subject separation has also been tackled with unsupervised learning solutions [18]. However, the main drawback of subject separation is that users are not identified, which is not suitable for use cases requiring personalized service provision.

Implicit data association

Implicit data association methods involve a data-driven model that implicitly learns user-specific features that may include personal habits, sensor signals captured from their personal belongings, and other relevant features that can be extracted from environmental data. Implicit data association has been achieved with multi-task learning and multi-label classification.

In the first case, each learning task consists of recognizing the activities of a specific user [70]. These methods typically rely on Hidden Markov Models (HMM), where, for instance, each user is assigned to a specific chain of hidden states [71, 72, 73, 17].

In the case of multi-label classification, a single unified learning task is considered to recognize the activities performed by different users [74, 75]. In this scenario, different techniques for multi-label classification have been explored by the HAR community, like binary relevance [76, 77], classifier chain [78], label combination [79, 72, 80, 81, 82, 83, 84, 85, 86], and random k-labelsets [87].

Overall, implicit data association methods assume that the available training set included all the possible combinations of activities that the involved users could have performed individually or collaboratively in the same living space.

Supervised data association

Supervised data association approaches consider data association as a separate learning problem before activity classification [19, 88, 18, 89, 90]. For instance, in [19], labeled data about behaviors and habits of the subjects of a smart environment are used to train a supervised classifier that attributes a subject to each sensor event. The main problems of supervised data association approaches are that (i) they require additional labeled data to train a data association model

and that (ii) such a model heavily relies on the specific environment and users' habits considered during training.

2.3.2 Labeled data scarcity in context-aware HAR

Another HAR application that aggravates the labeled data scarcity issue is the context-aware recognition of low-level activities based on mobile devices. In this scenario, contextual information about the users' surroundings (e.g., semantic location, speed, and weather conditions) is used to better discriminate activities with similar motion patterns like *standing* and *getting the elevator* [22]. However, in this application domain, it is not realistic to rely on supervised DL models. Indeed, they would require comprehensive training sets containing all the possible context conditions in which activities may be performed.

2.3.3 Mitigating labeled data scarcity in HAR

To mitigate the labeled data scarcity problem, the HAR research community investigated data augmentation, transfer learning, semi-supervised learning, and unsupervised learning (e.g., self-supervised learning) approaches [6].

Data augmentation is a popular solution to handle data scarcity, especially considering imbalanced datasets [91, 92]. These approaches generate new samples by slightly perturbing the available data, or by relying on generative AI solutions, like Generative Adversarial Networks (GANs) [93, 94]. However, it is questionable if such data augmentation techniques are effective when the original dataset is extremely small since they cannot fully compensate for the lack of diverse and representative training data.

Transfer learning methods usually take advantage of models trained on a source domain with a significant amount of labeled data. Such pre-trained models are then fine-tuned in a target domain using small amounts of labeled samples [95, 96, 97, 98].

On the other hand, semi-supervised approaches for HAR rely on small labeled datasets to initialize the model, which is then incrementally updated by leveraging the unlabeled data stream [10, 99, 100]. Semi-supervised methods for HAR include self-learning [101], co-learning [102], active learning [103, 104, 105, 106, 107],

and label propagation [108].

Additionally, unsupervised approaches have been exploited in different ways by the HAR community. For instance, in [109], unsupervised learning is used to derive activity clusters from unlabeled sensor data, requiring a few annotations to reliably associate activity labels to the clusters. On the other hand, in multi-subject smart environments, unsupervised methods have been explored also to perform subject separation [110, 111, 112].

Finally, among unsupervised methods, self-supervised learning strategies leverage large amounts of unlabeled data to pre-train a model capable of generating reliable feature representations of sensor data [113, 114, 115]. The pre-trained model is then fine-tuned using a limited amount of labeled data. More specifically, a surrogate objective (i.e., the pretext task) is designed so that optimizing it would lead the DL model to learn features that are meaningful also for the main classification task (i.e., the downstream task). After training the model to accomplish the pretext task, fully connected layers for classification can be added at the top of such a model before fine-tuning it for the downstream task.

NeSy methods to mitigate labeled data scarcity

As already discussed in Section 2.2.3, NeSy methods have the potential to improve the classifiers' recognition rates by infusing domain constraints into DL models. This may be especially true in data-scarce scenarios, where the activity classifiers would struggle to learn such constraints directly from data.

Moreover, NeSy methods could be potentially coupled with the other techniques previously presented in this section to further improve the recognition rates in data scarcity scenarios. For instance, in [1], a NeSy approach (not based on knowledge infusion) is combined with semi-supervised learning to maximize the recognition rates of a context-aware classifier in charge of recognizing low-level activities. As another example, we believe that domain constraints could be infused into a DL model during the fine-tuning phase of a self-supervised learning procedure to further minimize the amount of required labeled data.

2.4 The lack of interpretability issue

Another problem that limits the deployment of DL models for HAR is their opacity: it is challenging to understand the rationale behind their predictions [116]. Explainable Artificial Intelligence (XAI) approaches recently emerged to address this problem [117], by providing a human-understandable explanation associated with each model's prediction.

Important decisions in pervasive applications may rely on the output of a HAR classifier. Hence, inferring why a specific activity was predicted is essential to provide solutions that are understandable, trusted, and transparent [118]. For example, consider a healthcare system that analyzes the daily routines of elderly subjects. The detection of their activities is one of the fundamental steps to detect higher-level behaviors to support clinicians' diagnoses (e.g., cognitive decline) and interventions [119]. In such a scenario, XAI would allow clinicians to increase their trust in decision-support systems that rely on activity recognition. Explanations are also useful to data scientists who need to refine the recognition system by introducing, removing, or re-positioning sensors, modifying algorithms and system parameters, or revising/extending the training set. An explainable system would also make it possible to include the users in the loop, by showing them which activities are released to clinicians and how the system inferred their execution by the resident.

2.4.1 XAI taxonomy

According to DARPA[1] [120], there are three main categories of XAI approaches: *interpretable model* methods, *model induction* methods (also called *black box* methods), and *deep explanation* methods.

XAI *interpretable model* approaches are applicable to classic ML algorithms, like decision trees and Bayesian Rule Lists (BRL), that are inherently explainable [121, 122]. For instance, BRL models are built by learning from labeled data a set of human-readable probabilistic rules that correlate the input features with the target classes. These rules can be used both for classification and, at the same

[1]The Defense Advanced Research Projects Agency (DARPA) is a research agency of the United States Department of Defense responsible for the development of emerging technologies

time, to interpret the rationale behind each output. However, a major open issue in XAI is to explain complex models whose interpretation is more challenging, like the ones based on DL.

XAI *model induction* approaches like LIME [123] and SHAP [124] consider the classifier as a black box and correlate the input and the output to induce the explanations [125]. For instance, LIME generates each explanation by deriving a linear model based on the correlations between perturbed versions of the input and the predicted class. The weights of the resulting linear model indicate the most important features for classification. However, model induction approaches have been recently criticized because they can not reveal the hidden patterns captured by the black boxes during training, thus providing explanations that may not identify the actual reasons for the prediction [126].

Finally, XAI *deep explanation* methods have been proposed to derive explanations from deep learning models [127]. The most common approaches in this category are saliency-based methods, that analyze the activation of the neurons at intermediate layers of the network. For instance, Grad-CAM [128] analyzes the activation of the neurons in the last convolutional layer of the model (i.e., the ones that capture high-level information) to infer which portions of the input are important for the classification. However, recent studies indicate that those approaches may not reveal meaningful explanations [129]. More sophisticated deep explanation approaches introduce specialized layers in the network to learn some target class prototypes [130]. Intuitively, each prototype encodes a representative data sample of the training set for a specific target class. Each prediction is explained by showing the prototypes that are most similar to the input data. For instance, the work in [131] relies on metric learning to compute the distance from the input to the closest prototype, where a fixed number of prototypes is learned thanks to a specifically designed layer in the network.

2.4.2 Evaluating the effectiveness of explanations

A challenging problem in XAI is how to evaluate the effectiveness of explanations [120]. The choice of the evaluation strategy is strictly related to the goal of the underlying system. The target users may be (a) end-users who use AI in their

daily lives without knowledge about machine learning, (b) data scientists who use machine learning for analysis, or (c) experts in machine learning.

Moreover, depending on the target users, different aspects should be considered for evaluating the effectiveness of the explanation. In the literature, several metrics have been proposed [132], but not in the HAR domain. Most of them are based on directly interviewing the end-users. In the following, we report the most common ones:

- *Mental model.* This metric aims at assessing how a user understands the underlying system [133]. This metric is usually measured by explicitly asking the end-users their interpretation of the system's decision-making process, with the objective of evaluating the completeness of explanations.

- *Explanation usefulness and satisfaction.* This metric indicates the understandability and sufficiency of details in explanations [134]. This metric is usually measured in a qualitative or quantitative way through questionnaires.

- *User trust.* Trust is the cognitive factor that influences the perception of the system (positively or negatively) [135]. Trust and reliance are usually measured by asking the end-user opinions during and after the interaction with the system. Prior knowledge and beliefs can also influence the initial state of trust, which may change while interacting with the XAI system.

- *Computational Measures.* This category of metrics quantitatively evaluates the interpretability without involving the end-user. Indeed, reliance on human evaluation of explanations may lead to persuasive explanations rather than transparent systems due to user preference for simpler and intuitive explanations [136]. The goal of such approaches is to automatically compute the correctness, consistency, and fidelity of XAI methods [132].

2.4.3 XAI in activity recognition

XAI approaches have been mainly proposed for video-based activity recognition [137, 138, 139]. However, generating meaningful explanations for predictions based on sensor data is more challenging.

There exist only a few research efforts that focus on explainable approaches for sensor-based activity recognition and they consider only inherently interpretable models, like the one proposed in [140] that is based on the feature importance derived by the model parameters of classic ML methods. The authors in [141] rely on a rule-based classifier. During the training process, the model learns a set of human-readable rules that encode the correlations between sensor events and activities. The results indicate that the proposed model reaches recognition rates similar to well-known interpretable classifiers (e.g., Decision Tree, JRip) while generating significantly less complex rules. The work in [142] proposed a model based on fuzzy logic rules. The solutions proposed in these works do not generate explanations that are easily understandable by non-expert users. On the contrary, HealthXAI [143] provides explanations in natural language targeted to clinicians. However, that work focuses on the detection of high-level abnormal behaviors of elderly subjects in smart-home environments. Hence, the explanations in HealthXAI are derived from an underlying activity recognition classifier (i.e., a decision tree) that actually does not provide explanations.

The major limit of all of the above-mentioned works is that they are only XAI interpretable model approaches, that mainly do not tackle the problem of making explanations understandable also to non-expert users (e.g., clinicians, caregivers). Hence, it is still an open problem to understand if and how XAI can be combined with DL-based activity recognition with sensor data.

2.5 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 neuro-symbolic AI mitigate labeled data scarcity in multi-subject HAR applications?

Labeled data scarcity is one of the main issues that limit the deployment of supervised deep learning models for HAR in real-world applications. This problem

is emphasized in multi-subject HAR, where the existing literature mainly tackled data association in a data-driven way.

In Chapter 3, we propose a NeSy approach that relies on symbolic reasoning to perform data association without requiring any supplementary labeled data. After data association, an activity classifier is trained through a semi-supervised learning strategy based on active learning. Symbolic reasoning is hence used also to refine the predictions of such a model, thus further mitigating data scarcity and reducing the number of active learning queries triggered to the users.

Q2) Can neuro-symbolic AI based on knowledge infusion provide a more robust solution for addressing labeled data scarcity in context-aware HAR compared to existing approaches?

NeSy approaches have been already considered in the literature to mitigate data scarcity in context-aware HAR applications [1]. However, they only considered domain knowledge after the training process of the activity classifier, thus limiting the opportunity to unlock the full potential of Neuro-symbolic AI.

In Chapter 4, we introduce an innovative NeSy method that leverages Knowledge Infusion to mitigate data scarcity while being more robust than existing NeSy solutions in the presence of noisy context data.

Q3) How can we build knowledge infusion methods for context-aware HAR without the need for computationally expensive symbolic reasoning modules after deployment?

All the existing NeSy methods for HAR (including the one we present in Chapter 4) require computationally expensive symbolic reasoners after training the activity classifier. This could limit the deployment of such solutions on resource-constrained devices (e.g., mobile and wearable devices).

In Chapter 5, we present a novel Knowledge Infusion approach based on a semantic loss function that infuses domain knowledge into the activity classifier only during training, thus avoiding symbolic reasoning after deployment.

Q4) How can we measure the interpretability of DL models for HAR in order to assess the interpretability benefits produced by neuro-symbolic AI?

Neuro-symbolic AI may enhance the interpretability of deep learning models in the HAR domain. However, in this field, there is no quantitative metric to measure the interpretability level of activity classifiers based on deep learning. This is due to the fact that it is challenging to apply existing XAI methods to sensor data.

To address this problem, in Chapter 6, we introduce a novel methodology that enables the use of existing XAI techniques for sensor-based HAR. Additionally, we present the Explanation Score, a metric that measures the coherence of the explanations obtained through XAI methods with human knowledge about the HAR domain. Unfortunately, due to time constraints, we use the Explanation Score only to evaluate purely data-driven models. Nonetheless, we believe that this metric can be adopted in the future to evaluate whether Neuro-symbolic AI for HAR can make DL classifiers intrinsically more interpretable.

Chapter 3

Neuro-symbolic HAR in multi-subject smart-home environments

3.1 Introduction

The majority of existing recognition methods for high-level activities like Activities of Daily Living (ADLs) considered single-subject smart-home settings, where only one user lives in the environment [8]. However, it often happens that multiple users live in the same home (e.g., an elderly and a caregiver). In these settings, to accurately detect ADLs for the fragile target users, it is crucial to correctly discriminate the activities performed by each subject. Moreover, differently from single-subject settings, multiple users may perform ADLs jointly (e.g., Alice and Bob are cooking together) and concurrently (e.g., Alice watches TV while Bob is cooking).

Recently, several research efforts on multi-subject ADL recognition have been proposed in the literature [144]. The major open research problem in this area is that environmental sensors do not directly identify the users who generated sensor events (e.g., the opening of a kitchen drawer revealed by a magnetic sensor). Hence, to better recognize the activities performed by each user, it is crucial to perform *data association*: mapping each environmental sensor event to the user

which triggered it [16]. Existing multi-subject solutions assume the complete availability of labeled data to perform data association implicitly during model training [17] or as a separate supervised learning problem [19]. Hence, these approaches further aggravate the labeled data scarcity problem of deep learning classifiers.

In this chapter, we propose MICAR: a novel multi-subject activity recognition framework that combines semi-supervised learning with neuro-symbolic AI. Wearable and environmental sensor data are leveraged to derive high-level semantic information about the users (e.g., their posture and location in the home environment) to reliably perform data association using symbolic reasoning, thus avoiding any additional labeled data. Labeled data scarcity is further mitigated thanks to a novel cache-based active learning approach that continuously improves an activity classifier (initialized with limited labeled data) while triggering a limited number of questions. MICAR is capable of detecting both individual and group ADLs.

Our experiments on the MARBLE dataset [2] indicate that MICAR reaches a high recognition rate (F1 score ≈ 0.89) that is slightly behind a fully supervised approach while triggering a low number of active learning queries (query rate $\approx 3\%$). Moreover, our results confirm that our data association solution leads to a recognition rate that is only 2% behind the one obtained by an ideal approach based on ground truth. Our results also indicate that MICAR is accurate in detecting the number of users that jointly perform an ADL.

The rest of the chapter is organized as follows. Section 3.2 formally describes the multi-subject activity recognition problem. Section 3.3 describes the overall architecture of MICAR. Section 3.4 describes each component of MICAR in detail. Section 3.5 presents the evaluation methodology and the main results obtained on MARBLE [2], a dataset we recently published that we used to evaluate MICAR. Finally, Section 3.6 discusses some limitations of MICAR.

3.2 The data association problem

Given a limited amount of labeled data, the objective of the activity recognition system (named just *system* in the following) is to periodically infer for each user the activity of daily living (ADL) that she has been performing. The system also detects situations where ADLs are performed in cooperation by multiple users. Intuitively, a set of users is jointly performing an ADL when those users are in the same place and, according to the system predictions, they are performing the same ADL[1].

Let $\mathbf{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of users (the smart-home residents) and $\mathbf{A} = \{A_1, A_2, \ldots, A_k\}$ the set of target ADLs. Given an instant t, the system predicts for each user the activity prediction $\langle u, A, L, t\rangle$, where u is the user that performed activity A in the semantic location L. Hence, the system returns a set of tuples $PA_t = \{\langle (u_r, \ldots, u_s), A_i, L_j\rangle | \langle u, A_i, L_j, t\rangle \forall u \in (u_r, \ldots, u_s)\}$. Each tuple represents the set of users that jointly performed A_i the same ADL in the same semantic location L_j.

In order to achieve this goal, the system continuously analyzes a stream of time-stamped events coming from inertial and environmental sensors. Given an instant t and a user u, the system needs to solve a *data association problem* to derive a personalized stream $s(u)^t$ of sensor events associated with user u and collected in a time window $[t, t + k]$ where k is the window size parameter. For example, suppose that Anna opens the fridge door at time t'. The corresponding sensor event (and its timestamp) generated by the magnetic sensor connected to the fridge door and recorded by our system should be associated with Anna and hence considered part of $s(Anna)^t$ when $t \leq t' \leq t + k$.

The data association problem is straightforward for events coming from inertial sensors on personal devices but challenging for environmental sensors.

[1]Note that here we assume that users that perform the same ADL in the same semantic place at the same time are actually jointly performing the ADL. This is indeed the case in our considered setting.

3.3 MICAR's architecture

The general architecture of MICAR is depicted in Figure 3.1. Several environ-

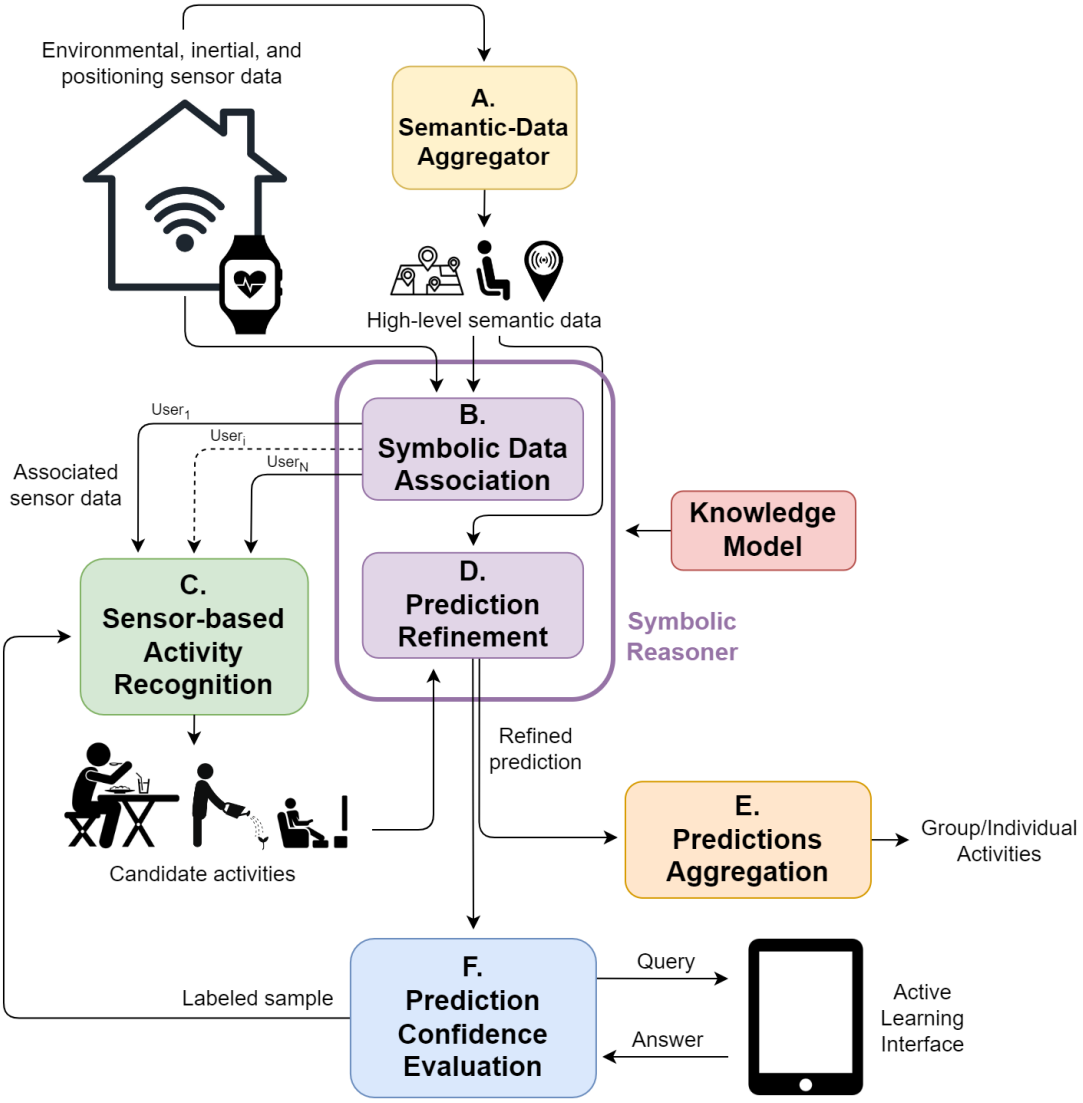


Figure 3.1: Overall architecture of MICAR

mental sensors are deployed in the home (e.g., plug sensors, magnetic sensors, motion sensors) to capture the interaction of the users with the surrounding environment. Moreover, each user wears a smartwatch that collects data from its inertial sensors (e.g., accelerometer) and a micro-localization system (e.g., BLE beacons, WiFi) deployed in the environment. Raw sensor data are continuously

transmitted to a smart home gateway, which is in charge of running the algorithms of MICAR.

First, the SEMANTIC-DATA AGGREGATOR module pre-processes sensor data to infer higher-level semantic information (i.e.,, users' locations and low-level activities, the position of the environmental sensors in the smart home). High-level semantic data, as well as raw sensor data, are then transmitted to the SYMBOLIC DATA ASSOCIATION module. This module relies on symbolic reasoning on high-level semantic information and sensor events to generate a personalized stream of inertial and environmental sensor data for each user. The rationale is that it is possible to use common-sense knowledge in the activity recognition domain to exploit high-level semantic data to derive the most likely correspondence between each environmental sensor event and the user that triggered it.

Each personalized stream is then processed by the SENSOR-BASED ACTIVITY RECOGNITION module. This module relies on an incremental semi-supervised classifier to detect the ADLs performed by a specific user. The output of the classifier is a probability distribution over the possible activities. The recognition model is initialized with a limited number of labeled data from a few users (e.g., 2 in our experiments) that in an initial phase contributed to a small labeled data acquisition campaign.

High-level semantic information is then processed again by the PREDICTION REFINEMENT module to refine the machine learning classification. Indeed, ADLs associated with a positive probability but in contrast with the current high-level semantic data (e.g., watching TV when the TV is not turned on) are removed from the probability distribution.

The PREDICTIONS AGGREGATION module combines the refined predictions from each user to output both individual and joint activities performed by the residents. In particular, a heuristic method determines whether multiple users are performing the same activity.

In parallel to PREDICTIONS AGGREGATION, the PREDICTION CONFIDENCE EVALUATION module evaluates the uncertainty of the refined prediction. If the uncertainty is greater than a threshold, an active learning process is started: the system triggers a query to ask the user which activity she is performing through a dedicated interface. The feedback is used to update the incremental activity

recognition classifier. Our active learning method is based on a cache to reduce the number of triggered questions.

In the next section, we describe each component of MICAR in detail.

3.4 MICAR under the hood

3.4.1 Sensing sources

The users are monitored with a combination of wearable and environmental sensors. In particular, each user wears a smartwatch, equipped with inertial sensors (i.e., accelerometers, gyroscopes, and magnetometers) to track her physical movements. Inertial sensors are particularly useful for capturing ADLs that are characterized by specific gestures (e.g., washing dishes). Smartwatches also collect data (e.g., RSSI) from a positioning system deployed in the home (e.g., BLE beacons, Ultra-Wideband, WiFi access points). Positioning data is particularly useful to continuously monitor the semantic position of the user. Since the smartwatch is a personal device, the collected data can be automatically associated with the resident's identity.

Environmental sensors capture the interaction of the residents with the home infrastructure. For example, magnetic sensors detect the opening and closing events of doors and drawers, pressure mats on chairs reveal if someone is sitting, and smart plugs detect the usage of home appliances. As we already mentioned, environmental sensors cannot identify the resident which triggers them since they only output their status.

3.4.2 Semantic-data aggregation

The SEMANTIC-DATA AGGREGATOR module receives the raw data from the sensing sources described above. The objective of this module is to derive higher-level semantic information. As we described in Section 3.3, MICAR uses high-level semantic information to compute data association as well as to refine the classifier's prediction.

The SEMANTIC-DATA AGGREGATOR module derives the *personalized context*

for each user and the *home context* for the home environment. Given a time instant t, the *personalized context* of a user u is denoted with $C(u)^t = (l(u)^t, p(u)^t)$, where $l(u)^t$ is the location of u in the home at time t and $p(u)^t$ is the posture of u at time t. For instance, if Bob is sitting in the kitchen at time t then $C(Bob)^t = (kitchen, sitting)$. On the other hand, the *home context* C_H^t encodes the status and the position of each sensor in the home. In the following, we describe how $C(u)$ and C_H are computed from raw sensor data.

User's semantic position

In the following, we describe how we derive the semantic position $l(u)^t$ of a user u at time t. In our implementation, the smartwatch is in charge of collecting RSSI data from a positioning infrastructure composed of a combination of BLE beacons and WiFi access points. Raw RSSI data are segmented with a sliding window of size n_l and overlap p_l. Then, we apply a Savitzky-Golay filter to smooth raw RSSI data. In our experimental setup, we use $n_l = 5s$ and $p_l = 50\%$.

For each temporal window, we extract a feature vector, where each feature encodes the mean RSSI signal of the window from a specific source (i.e., a specific BLE beacon or WiFi access point). In our experimental setup, the *mean* was sufficient to characterize each signal, while the use of other statistical properties did not lead to any improvement in the positioning accuracy. Finally, a machine learning classifier is in charge of classifying the semantic position of the user from the feature vectors. In our experiments, we used a Random Forest classifier.

Note that the organization of the home in semantic positions should be performed in an offline phase, and its granularity depends on the accuracy of the underlying micro-localization system. A coarse granularity may consider room-level semantic positions (e.g., living room, kitchen, dining room), while a fine-grained granularity may map specific regions of each room into semantic positions (e.g., cooking area, dining table, and sink area).

In our experimental setup, we implemented a micro-localization infrastructure at room-level granularity based on a combination of 5 BLE beacons[2] uniformly installed within our smart-home lab and 26 WiFi access points that could be

[2]We performed several experiments considering up to 10 BLE beacons, but we observed interference problems when considering more than 5 beacons

detected in its surroundings. Our infrastructure reaches an average positioning error of $1-2$ meters. However, we did not consider these results to be satisfactory for an accurate data association.

In the literature, several solutions have been proposed for more accurate indoor positioning [145]. MICAR is agnostic to the specific micro-localization system being used, and we preferred to use ground truth information about positioning data in our experiments, in order to focus on multi-subject activity recognition only. We expect that new technologies (e.g., UWB) will be significantly more accurate in indoor localization, and MICAR could adopt them to perform reliable data association.

User's posture

The posture $p(u)^t$ (e.g., standing, sitting, lying) of a user u at time t is derived by feeding a machine learning classifier with the inertial sensors data from the smartwatch. First, we pre-process raw data by applying a median filter to reduce the noise. Then, we apply sliding window segmentation, with a window size of n_p seconds windows and overlap p_p. In our experimental setup, we use $n_p = 8s$ and $p_p = 80\%$. For each temporal window, we extract several features that are well-known to be accurate for low-level activity recognition [3]. We obtain in total 120 inertial features, which are then dimensionally reduced to d_p values through the ANOVA technique [146], and finally standardized. In our experiments, we determined $d_p = 84$. Each feature vector is provided to a machine learning classifier to distinguish between different postures. In our experiments, we used a simple multilayer perceptron (MLP) to discriminate between *sitting* and *not sitting*. Note that this process is model-agnostic. Hence, if the set of user postures to be recognized requires more powerful solutions, it will be possible to introduce a more complex deep neural network in charge of automatically extracting meaningful features from raw sensor measurements.

Sensor status and position

As we previously mentioned, MICAR also computes C_H^t as the context of the home environment. An important contextual aspect is the semantic position of

each sensor, which we consider as prior knowledge defined during the deployment phase in the smart home. During the deployment phase, we also map each environmental sensor to a semantic concept. For instance, when the magnetic sensor installed on the fridge door fires, it generates the high-level event (*fridge_door*, *kitchen*, *OPEN*), which means that the fridge door in the kitchen has been opened.

C_H^t keeps track of the current status of environmental sensors by considering the previously mentioned high-level information.

Example 3.1 *Consider a home H equipped with two plug sensors: one to detect the usage of the electrical stove in the kitchen and one to detect the usage of the television in the living room. Suppose that at time t Bob is watching TV and that no one is using the electrical stove. In this case* $C_H^t = \{$(*stove,kitchen,OFF*), (*television*, *living_room*, *ON*)$\}$.

3.4.3 Symbolic data association

Given the high-level semantic data from the SEMANTIC-DATA AGGREGATOR and the raw sensor data collected from inertial and environmental sensors, the goal of data association is to periodically compute for each user u a personalized sensor data stream $s(u)^t$. A stream $s(u)^t$ consists of the inertial sensor readings gathered from the personal device of u, and the environmental sensor events triggered by u in a time window $[t, t + k]$, where k is the size of the segmentation window. Note that $s(u)^t$ is computed every time a new environmental sensor event (e, st, t) occurs. In our experiments, we empirically determined $k = 14s$.

As we previously mentioned, the challenge of data association is to assign environmental sensor events to the user that most likely triggered it. Indeed, an environmental sensor event (e, st, t) (e.g. (*fridge_door*, *OPEN*, *12:32*)) cannot directly identify the user who triggered it.

MICAR performs data association by exploiting the high-level semantic data. In particular, it approximates a stream $s(u)^t$ by including all the environmental events that are *consistent* with $C(u)^t$ and C_H^t. The notion of *consistency* is inherently related to the semantics of the context and the action revealed by the event. The SYMBOLIC DATA ASSOCIATION module of MICAR is implemented

with ontological reasoning. In particular, an OWL2 ontology defines the relationships between environmental sensor events and high-level semantic information. In the following, we describe some axioms that we encode in our ontology.

Among other constraints, our ontology imposes that a user can trigger a sensor event only if she is in the same *semantic position* where the sensor is located (e.g., Alice cannot turn on the TV in the living room while she is in the bedroom). Other axioms combine *user's posture* and *sensor status and position* to better associate environmental sensor events when multiple users are in the same semantic position at the same time. For instance, the activation of the pressure mat can be associated only with those users who recently switched to the *sitting* posture. Similarly, the *sitting* posture is not compatible with sensor events that can be triggered only while *standing* (e.g., turning on the stove).

In general, when a sensor event (e, st, t') is triggered, our system checks its semantic consistency for each user u using ontological reasoning. In particular, MICAR adds factual observations to the ontology to describe the sensor event, the context $C(u)^{t'}$, and the context C_H. Then, by using the automatic consistency check of the resulting ontology, the system decides whether (e, st, t') should be included in $s(u)^t$ (with t' in the time window defined by t)

The output of the SYMBOLIC DATA ASSOCIATION module is hence a personal stream $s(u)^t \ \forall u \in \mathbf{U}$. The solution is approximate since there may not be sufficient information to associate an event to a single user and, in this case, the event will be associated with the stream of each candidate user.

Example 3.2 *Suppose that Anna and Bob are both in the kitchen, and the magnetic sensor on the fridge generates an event at time t, thus indicating that someone opened it. Suppose that Anna is standing, while Bob is sitting on a chair. This semantic information is detected by the* SEMANTIC-DATA AGGREGATOR *module. Hence, MICAR adds to the ontology the observations about the users in the home (i.e., Anna and Bob), their high-level semantic information (i.e., Anna is standing in the kitchen, Bob is sitting in the kitchen), and the triggered environmental sensor event (the fridge magnetic sensor is ON). By performing a consistency test, our ontology derives that the opening fridge event is consistent with Anna's context, while it is not consistent with Bob's context (i.e.,*

a user cannot open the fridge if he is sitting*). Hence, in this case, the fridge event will be included in* $s(Anna)^t$ *and not in* $s(Bob)^t$*.*

We show a small sample of our ontology in Figure 3.2. In order to simplify the visualization, the ontology is represented as a graph where each node is an entity, while each edge encodes a relationship.

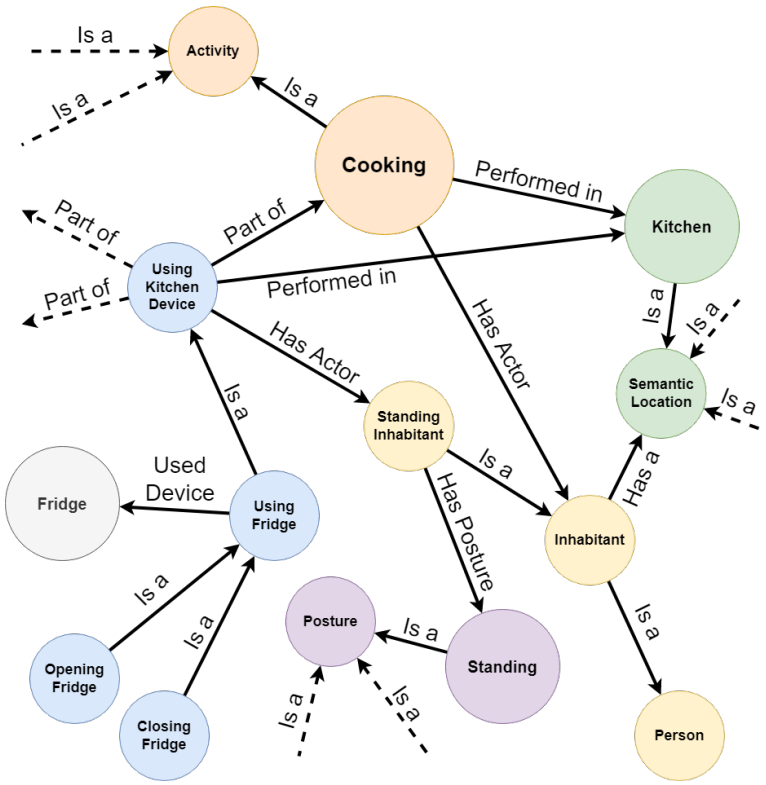


Figure 3.2: A simplified representation of a small portion of our ontology. Each node encodes an entity, while each edge encodes a relationship.

3.4.4 Sensor-based activity recognition

The objective of the SENSOR-BASED ACTIVITY RECOGNITION module is to infer the activities performed by each user in the home. For each user u, it periodically processes the personalized stream $s(u)^t$ received from the SYMBOLIC DATA ASSOCIATION module to derive the activity performed for u at time t. Since the focus of this work is on performing data association without any labeled sample,

the SENSOR-BASED ACTIVITY RECOGNITION module relies on a pipeline for standard machine learning methods that we already adopted in other works within our research lab. Since MICAR is agnostic to the activity classifier, this model can be easily replaced with more powerful deep-learning solutions if required.

Segmentation and feature extraction

MICAR considers each personalized $s(u)^t$ as a temporal window of size k. In order to improve the recognition model, we also compute overlapping segmentation between consecutive windows considering an overlap factor of p_{ar}. In our experiments, we determined $p_{ar} = 80\%$.

From each segmentation window, MICAR extracts different features from inertial and environmental sensor data. Considering inertial data, we apply a median filter for noise reduction. Hence, we extract 120 well-known statistical features from accelerometer, gyroscope, and magnetometer data [3]. Examples of such features are: *root mean square*, *kurtosis*, *symmetry*, *zero-crossing rate*, *number of peaks*, and *energy*, and the *pearson correlation*. Considering environmental sensor data, we extract 36 features that are based on the status of the smart-home sensors and the number of their activation and deactivation events. In particular, MICAR implements the feature extraction technique based on temporal decay that was proposed in [38]. In our experiments, we applied ANOVA to reduce the dimensionality, reducing the feature space from 156 features to 84.

Activity recognition

Each feature vector fv generated from the personalized stream of a user u is provided to an incremental single-inhabitant ADLs classifier h to derive the probability distribution over the possible ADLs performed by u:

$$h(fv) = \langle p_{A_1}, p_{A_2}, \ldots, p_{A_n} \rangle$$

where $p_{A_i} \in [0, 1]\ \forall i$, $\sum_{i=1}^{n} p_{A_i} = 1$, and p_{A_i} is the probability $P(A_i|fv)$ that the user u is performing activity $A_i \in \mathbf{A}$, based on fv. Note that the activity recognition classifier is initialized using a limited amount of labeled data from a restricted

number of users. MICAR does not impose a specific choice for the single-subject classifier. In our experiments, we implemented a small neural network.

3.4.5 Prediction refinement

Activity recognition classifiers are sometimes not accurate, confusing ADLs that share similar sensor patterns. Considering machine learning-based approaches, the training set is often limited and it may not generalize on unseen activity patterns. As a drawback, the classifier can potentially derive a wrong activity. However, common-sense knowledge about the relationships between activities and high-level semantic information can be used to mitigate those classification mistakes.

MICAR uses the high-level semantic information $C(u)^t$ and C_H^t, computed by the SEMANTIC-DATA AGGREGATOR module to refine each activity prediction $h(fv)$. In particular, MICAR adopts an approach inspired by the one proposed in [1] (named *context refinement*). The PREDICTION REFINEMENT module of MICAR applies symbolic reasoning on high-level semantic data to exclude from the probability distribution predicted by the classifier those activities that are not consistent with the current high-level semantic information. In our experimental setup, this mechanism is based on the same ontology used by the SYMBOLIC DATA ASSOCIATION module.

Indeed, as it is possible to observe in Figure 3.2, our ontology also contains axioms about the relationships between high-level semantic data and activities.

MICAR evaluates whether an activity A is consistent by adding to the ontology the factual observations about the current high-level semantic information $C(u)^t$ and C_H^t and the fact that u is currently performing activity A. Inconsistent activities are removed from the probability distribution $h(fv)$, thus generating a refined probability distribution $h'(fv)$ over the remaining activities.

Example 3.3 *Suppose that MICAR inferred that Alice is watching television with* 60% *of probability, eating with the* 30% *of probability, and setting up the table with the remaining* 10%*. According to our ontology, the watching television activity can be carried out only when: a) the television is in the same user semantic position (user and sensor position), and b) the television is turned on*

(sensor status). Suppose that Alice is sitting at the dining table in the kitchen while eating, while the television in the living room is turned on. Hence, watching television is not consistent for Alice considering how this activity is described in our knowledge model. The resulting re-normalized probability distribution of Alice in this case is 75% *eating and* 25% *setting up the table.*

3.4.6 Predictions aggregation

The goal of the PREDICTIONS AGGREGATION module is to detect activities that are jointly performed by multiple users. For the sake of this work, we assume that a group activity occurs when two or more users perform the same activity A in the same smart-home location l during the same time interval.

Note that this module covers the case where different users start to perform the group activity at different times. For instance, consider a scenario where Alice watches the television and then eats, while Bob sets up the table and then eats. Bob starts eating 5 minutes before Alice. The PREDICTIONS AGGREGATION module would detect the group activity *eating* only when both Alice and Bob are eating. Moreover, the assumption on the semantic locations allows MICAR to capture the scenario where the same type of ADL is performed by different users in different rooms (e.g., Alice is watching TV in the living room, while Bob is watching TV in the bedroom).

In order to derive group ADLs, MICAR analyses the activities predicted for each user by the single-subject classifier and the users' location during their execution. In particular, for each user, the output of the classifier is processed in real-time to keep track of *stable activities predictions*. Given a user u, a *stable prediction* $S(u, A, L, [ti, tj])$ is generated from a sequence of consecutive feature vectors of u classified with the same activity A performed in the location l during the time interval $[ti, tj]$. In order to be considered stable, during $[ti, tj]$ the confidence on A should be higher than a threshold c for at least t times. In our experiments, we empirically determined $c = 0.75$ and $t = 3$.

Two users u_i and u_j jointly perform an activity A if there exists two stable predictions $S(u_1, A, L, [t_i, t_j])$ and $S(u_2, A, L, [t_l, t_k])$ such that $[t_i, t_j]$ and $[t_l, t_k]$ temporally overlap. The overlap between the time intervals determines the du-

ration of the joint activity. Clearly, this process works in a similar way for more than two users.

Note that the specific aggregation approach that should be adopted depends on the nature of the dataset and the specific target application. For instance, in a real-world scenario, more users could perform a collaborative activity while playing the same online multiplayer video game in different locations of the smart home, using different computers. For the sake of this work, we only target group activities that occur in the same location.

3.4.7 Prediction confidence evaluation

While MICAR uses symbolic prediction refinement to mitigate classification errors, the system may still be uncertain about the refined prediction. The PREDICTION CONFIDENCE EVALUATION module takes advantage of a semi-supervised strategy based on active learning to trigger a query to the user when the confidence in the refined prediction is below a certain threshold. Since this evaluation is performed on the output of a single-subject classifier, our active learning strategy is not targeted to joint activities.

For each $h'(fv)$ generated by the PREDICTION REFINEMENT module, we compute uncertainty based on the entropy of the probability distribution:

$$H(h'(fv)) = \sum_i p'_{A_i} \log \frac{1}{p'_{A_i}}$$

where p'_{A_i} is the refined probability distribution related to activity A_i. Note that the entropy measure is commonly used to compute the uncertainty in active learning [147]. When the entropy is higher than a threshold π, we assume that the system is uncertain about the activity currently performed by u. Hence, an active learning process is started, and MICAR asks to u feedback about the activity she was actually performing. For the sake of usability, only a few alternatives among the most likely activities are proposed. In our experiments, we determined $\pi = 0.6$

The feedback is then considered to update the incremental activity recognition classifier as a newly labeled data sample. MICAR updates the classifier when a

batch of w feedback is obtained by the users. In our experiments, we empirically determined $w = 32$ to balance the trade-off between convergence rapidity and recognition stability. For the sake of this work, the feedback from each user contributes to updating the same single-subject classifier that is used for every resident.

Active learning generally leads to good recognition rates for activity recognition [1]. However, a high number of queries negatively impacts the user experience. Since we periodically update the model with a batch of feedback, MICAR can potentially maintain the same uncertainty for consecutive feature vectors until the model is not updated. In order to mitigate this problem, MICAR implements a novel active learning strategy based on caching. In particular, for each user u, MICAR stores the latest uncertainty prompted to u as the set of the two most likely activities $\{A_i, A_j\}$ in the probability distribution[3], and the feedback provided by u. Hence, if the same uncertainty occurs multiple times within a short time period for a specific user, MICAR does not trigger additional queries and it uses the last feedback provided by u to update the classifier. When a new uncertainty occurs, MICAR overwrites the user's cache. After a certain amount of time, defined by the constant *CACHE_TTL*, MICAR invalids the cache. The MICAR's active learning approach is described in detail in Algorithm 1.

3.5 Experimental evaluation

3.5.1 The MARBLE dataset

In order to adequately evaluate MICAR, we collected a novel dataset (called MARBLE) in our smart-home lab. This dataset is publicly available [2]. To the best of our knowledge, there are no other publicly available multi-subject ADLs datasets that combine wearable and environmental sensor data to provide the high-level semantic information required by MICAR.

Due to privacy concerns, we were not able to acquire long-term data from actual users in real homes. Nonetheless, based on our previous experience in real-world deployments and in-the-lab data collections [148], we designed a new

[3]Note that we consider a set since the order of the two most likely activity is not relevant.

Algorithm 1 Cache-based active learning

$cache \leftarrow \emptyset$
$lastFeedback \leftarrow nil$
$t_{cache} \leftarrow nil$
$needToPromptUser \leftarrow$ True
for each feature vector fv of a user u generated at time t **do**
    $h'(fv) \leftarrow$ refined prediction from PREDICTION REFINEMENT
    if $H(h'(fv)) > \pi$ **then**
        $\{A_i, A_j\} \leftarrow$ the two most likely activity in the prediction
        if $\{A_i, A_j\} \neq cache$ OR $t - t_{cache} > CACHE\_TTL$ **then**
            $cache \leftarrow \emptyset$
            $needToPromptUser \leftarrow$ True
        end if
        if $needToPromptUser$ **then**
            Query prompted to user u with uncertainty A_i, A_j
            $A^{fv} \leftarrow$ the user feedback at time t^F
            $lastFeedback \leftarrow A^{fv}$
            $cache \leftarrow A_i, A_j$
            $t_{cache} \leftarrow t^F$
            $needToPromptUser \leftarrow$ False
        else
            $A^{fv} \leftarrow lastFeedback$
        end if
        Consider the feedback A^{fv} to update the model
    end if
end for

multi-subject dataset acquisition campaign in a smart-home lab with significant efforts in making it realistic and diverse. Moreover, the provided annotations are complete and very accurate. As depicted in Figure 3.3, we equipped the smart-

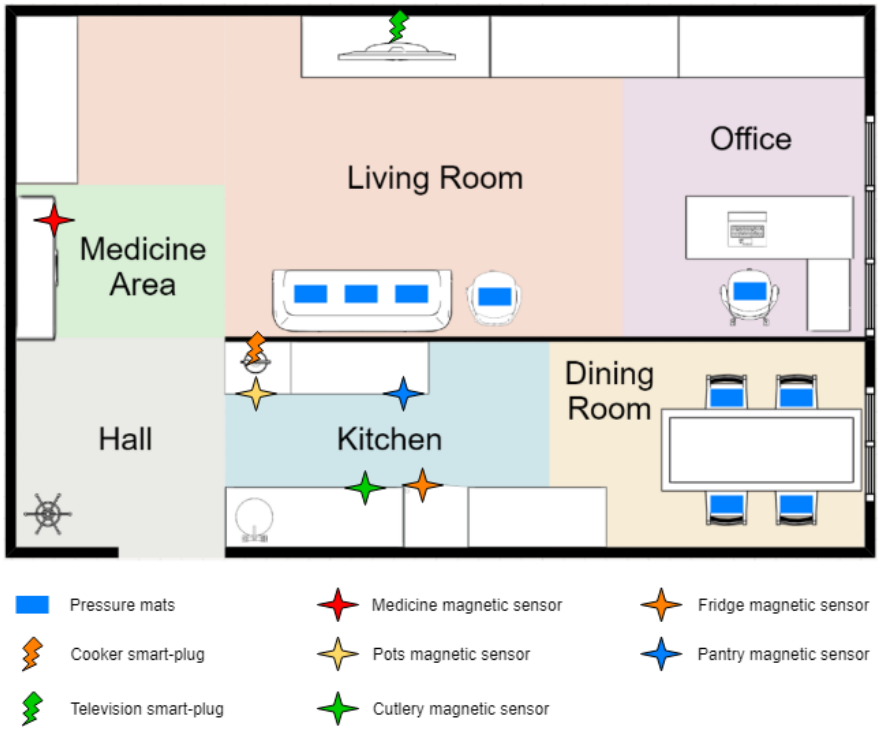


Figure 3.3: The simulated smart home during the data collection process of MARBLE

home lab with several environmental sensors: magnetic sensors to detect the opening and closing events of drawers (e.g., fridge, medicine cabinet), pressure mat sensors to detect when residents are sitting on chairs/sofa, and plug sensors to detect the usage of home appliances (e.g., TV, electric cooker). To monitor phone call activities, the residents carried an Android smartphone in their pockets running a dedicated application to detect starting and ending events of incoming and outgoing phone calls. The residents were also wearing a smartwatch[4] to collect data from inertial sensors (i.e., accelerometer, gyroscope, and magnetometer). We also deployed a positioning infrastructure composed of BLE beacons and WiFi APs. However, since indoor positioning is orthogonal to ADLs

[4]We used Huawei Sport 2 and other brands with similar features.

recognition, MARBLE only includes the ground truth about the semantic areas of each subject within the smart home. More specifically, the smart-home lab was divided into 6 semantic locations: *dining room*, *hall*, *kitchen*, *living room*, *medicine area*, and *office*.

MARBLE includes 13 ADLs: *answering phone*, *clearing table*, *cooking/cooking a hot meal*, *eating*, *getting in/entering home*, *getting out/leaving home*, *making a phone call*, *preparing/cooking a cold meal*, *setting up table*, *taking medicines*, *working/using PC*, *washing dishes*, and *watching TV*. We recruited 12 volunteers not involved in our research lab. We instructed the volunteers about the sequence of activities they had to perform, but they were free to execute them in their own way to increase the dataset variability. Our research team performed the annotations in real time, thanks to cameras. We designed four single-subject scenarios, three different scenarios involving two subjects concurrently performing both independent and joint activities, and four different scenarios of ADLs concurrently performed by four subjects. For instance, Table 3.1 shows one of the 2-subject scenarios that we designed. In this table, the flow of time is represented vertically, from top to bottom. Horizontal dashed lines indicate transitions between subsequent activities. When subjects collaboratively perform an activity the vertical line is suppressed. Each designed scenario is identified by a letter followed by the number of subjects involved during the data acquisition for that scenario.

Each scenario was repeated several times by different volunteers. Overall, we acquired 12 instances of 4 single-subject scenarios, 10 instances of 3 scenarios involving 2 subjects, and 10 other instances of 4 scenarios with 4 subjects involved. Table 3.2 shows, for each ADL type, the amount of recorded labeled data in minutes, and the average duration in seconds, while Table 3.3 shows the recorded time and the average duration of single-, 2-, and 4-subject scenarios. Note that, since we had time restrictions for data collection (due to the availability of volunteers), the execution time of each ADL was limited to a duration that in some cases does not reflect the actual time a person would need, but long enough to collect a significant amount of data. For instance, considering activities like *eating* or *cooking*, we asked our volunteers to perform them only for a few minutes.

Table 3.1: A scenario involving two subjects

A2

	Subject 1	Subject 2
morning	set table	cook
	eat	
	clear table	wash dishes
	use pc	watch tv make call
	watch tv	
	answer call	take meds
afternoon	prepare meal	cook make call
	take meds	set table
	eat	
	use pc make call	clear table
	leave home	
evening	enter home	
	eat	
	eat	answer call
	take meds	use pc
	make call	take meds
	watch tv	

Table 3.2: Statistics on labeled activities

	recorded minutes	average duration (s)	instances
ANSWERING PHONE	68.6	67.5	61
CLEARING TABLE	38.5	39.9	58
COOKING	80.5	81.9	59
EATING	150.2	28.2	320
ENTERING HOME	19.3	12.2	95
LEAVING HOME	13.7	16.1	51
MAKING PHONE CALL	63.6	53.8	71
PREPARING COLD MEAL	53.0	59.9	53
SETTING UP TABLE	53.9	39.4	82
TAKING MEDICINES	36.3	28.3	77
TRANSITION	276.1	12.9	1282
USING PC	94.1	86.9	65
WASHING DISHES	54.6	48.2	68
WATCHING TV	267.6	90.2	178

Table 3.3: Statistics on scripted scenarios

type of scenarios	recorded minutes	average duration (min)
single-subject	307.5	25.6 ± 4.0
2-subject	315.5	31.5 ± 7.7
4-subject	84.0	8.4 ± 1.8

3.5.2 Evaluation methodology

In the following, we describe how we evaluate the recognition rate of MICAR. Since our semi-supervised activity recognition classifier is incremental, we adopt a well-known evaluation technique for stream learning algorithms [149]. We pre-train the classifier using labeled data from 2 subjects that only contributed to single-subject scenarios. We use the remaining data to evaluate the evolution of the recognition rate and the number of questions triggered by active learning. We iterate over each data sample (i.e., feature vector), providing the classifier with one instance of a scenario at a time. Within a scenario instance, the order of data samples provided to the classifier reflects the temporal order of data collection.

Each data sample is first classified using the current model. The ground truth and the classification output are stored for evaluation. Then, we apply the active learning strategy presented in Section 3.4.7 to determine if the query is needed. If this is the case, we use the data sample labeled with the ground truth to update the recognition model and we update the number of triggered questions.

In order to show the evolution of the classifier, we use a sliding window approach to periodically compute both the overall F1 score and the percentage of triggered questions. Each window contains 800 data samples, and we consider an overlap factor of 75%.

In order to achieve statistically robust results, the whole experiment is repeated 100 times, averaging the results. Moreover, at each repetition, we also randomly shuffle the order of the scenario instances that we provide to the classifier.

3.5.3 Results

Recognition rate

In the following, we show results about the recognition rate of the SENSOR-BASED ACTIVITY RECOGNITION module of MICAR, including the prediction refinement step. Figure 3.4 depicts the evolution of the recognition rate using the evaluation methodology presented above. Thanks to active learning, the recognition rate quickly converges to high values. Without active learning, the classifier (only pre-trained using data from 2 users) is never updated, and the F1 score is stable on low values.

Figure 3.5 compares the recognition rate reached by MICAR with the one obtained by a supervised version of MICAR (i.e., with full availability of labeled data and without active learning). We will refer to this approach as *Supervised MICAR*.

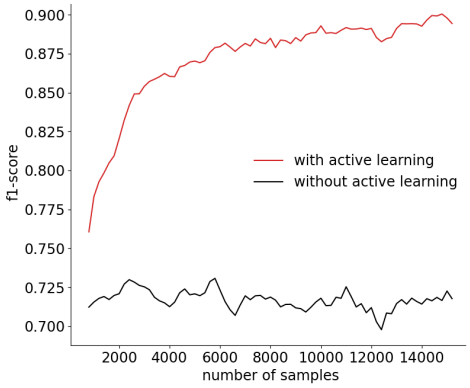


Figure 3.4: Evolution of the recognition rate of MICAR

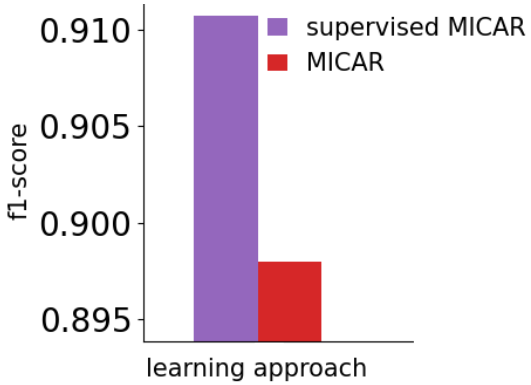


Figure 3.5: MICAR vs a fully supervised approach

We computed the F1 score of MICAR by considering the mean of the F1 scores obtained on the last four windows (see Figure 3.4). On the other hand, we computed the F1 score of *Supervised MICAR* using a leave-one-scenario-out cross-validation approach. At each fold, we considered a specific instance of a scenario as the test set, while the data of all the remaining scenario instances as the training set. To make our validation robust, we also removed from the training set: 1) data related to the other instances of the same scenario in the test set, and 2) data of the subjects in the test set. We observed that the recognition rate

of MICAR is only ≈ 1% behind the one reached by *Supervised MICAR*, with the great advantage of requiring a limited amount of labeled data. In Section 3.5.3 we show results about the number of active learning queries triggered by MICAR.

Figure 3.6 shows the confusion matrix generated by MICAR. Activities like

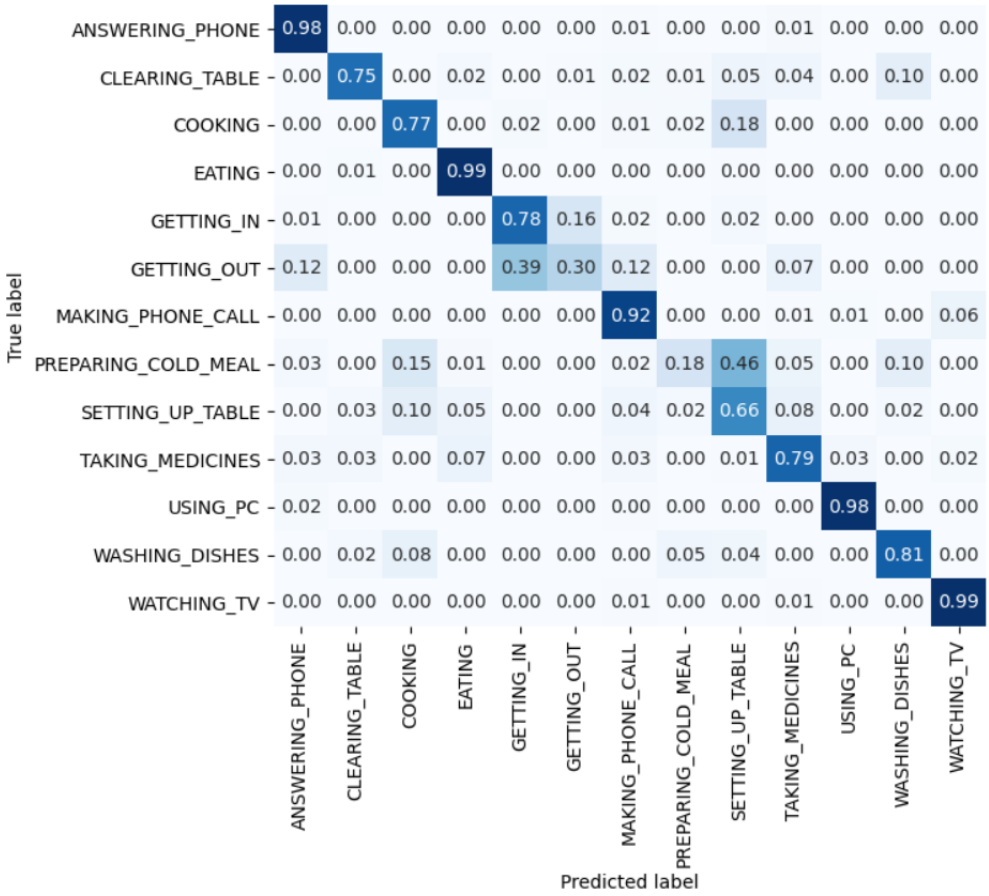


Figure 3.6: Confusion matrix

watching TV and *using PC* are recognized with a recall of around 98%. Indeed, in the MARBLE dataset, these activities are associated with specific semantic areas and environmental sensors that uniquely characterize them. For example, *watching TV* can only be performed in the *living room* triggering the smart plug sensor connected to the television.

On the other hand, those activities that are not uniquely characterized by available high-level semantic data exhibit a lower recognition rate. For example, the activities that can be performed by standing in the kitchen (e.g., *preparing a*

cold meal, *setting up the table*, and *cooking*) are often confused between them since they trigger similar sensors. Nonetheless, *cooking* still reaches good recognition rates thanks to the plug sensor that detects the electrical stove usage. Also, we observed that *washing dishes* is well-recognized even if it is associated with high-level semantic information similar to the above-mentioned kitchen-based activities. This is likely due to the ability of inertial sensor data to capture the gestures that uniquely characterize the activity. MICAR also confuses *getting in* and *getting out* activities due to their similar patterns. The remaining activities are well-recognized by MICAR.

Effectiveness of active learning

Besides the recognition rate, a fundamental aspect is the number of questions triggered by active learning due to its direct impact on user experience. Figure 3.8 shows that the percentage of active learning questions quickly converges to low values (below 5%) with a decreasing trend (i.e., the system asks fewer and fewer questions over time).

Figures 3.7 and 3.8 also compare our cache-based approach described in Section 3.4.7 with respect to a traditional method that does not use a cache (i.e., a query is triggered every time there is an uncertainty). We observed that the

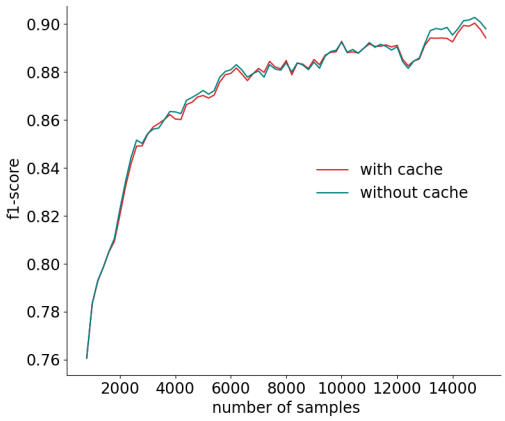


Figure 3.7: Impact of the cache on the evolution of the recognition rate

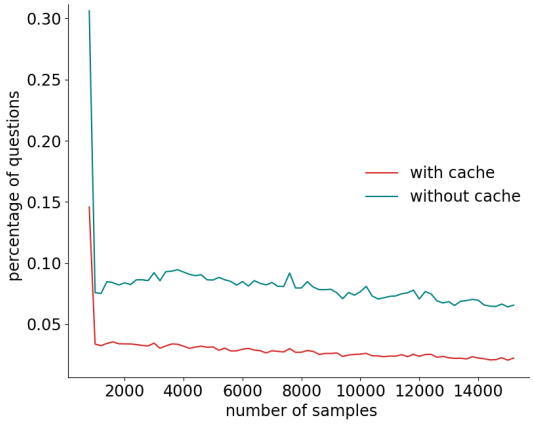


Figure 3.8: Impact of the cache on the evolution of the percentage of active learning queries

recognition rate of our cache-based method is almost identical to the one reached

by a traditional approach, while the percentage of questions is dramatically lower. We also observed that the cache was used by MICAR 66% of the times there was an uncertainty. This is due to the fact that MICAR updates the classifier with a batch-based approach. Hence, since the model update is delayed, the classifier often has the same uncertainty on consecutive feature vectors.

Symbolic data association

Figures 3.9 and 3.10 show the effectiveness of our data association method compared with two alternatives. The first is called *naive data association*, and it simply assigns each environmental sensor event to every user in the home, independently from high-level semantic data. The second one is called *perfect data association*, and it assigns each environmental sensor event to the correct user by using the ground truth. Clearly, *perfect data association* is an ideal approach that cannot be implemented in practice, and we consider it as an upper bound. Note that, to better highlight the impact of data association, we show the results that we obtained without PREDICTION REFINEMENT. The data association

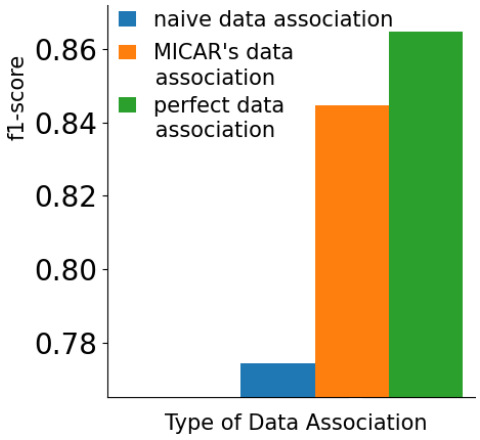


Figure 3.9: The recognition rate obtained by our data association approach with respect to a naive solution and an ideal solution

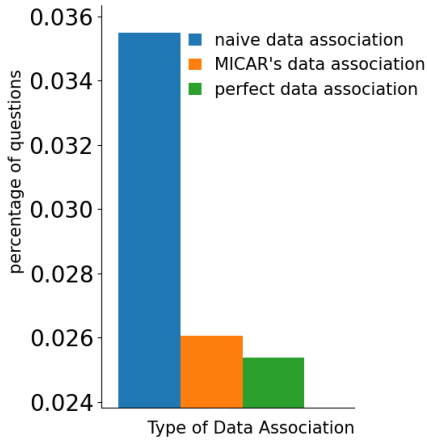


Figure 3.10: The percentage of questions obtained by our data association approach with respect to a naive solution and an ideal solution

strategy of MICAR significantly outperforms in terms of F1 score the *naive data association* approach (+6%). At the same time, our solution is only 2% behind a

perfect data association, without requiring any labeled data sample. These results suggest that our data association approach is accurate. Considering the number of active learning queries, the data association strategy of MICAR triggers a reduced number of queries than the *naive data association* solution, reaching very close results to *perfect data association*.

Prediction refinement

Figures 3.11 and 3.12 show the impact of the PREDICTION REFINEMENT module in refining the classification mistakes. We compare our method with two alternatives: *without prediction refinement* and *context as features*. The first is MICAR without the PREDICTION REFINEMENT module. Hence, the classification output is not refined using high-level semantic information. On the other hand, the *context as features* approach considers high-level semantic information as additional features in the machine learning process, instead of processing them with a symbolic approach after classification. Our results show that high-level semantic

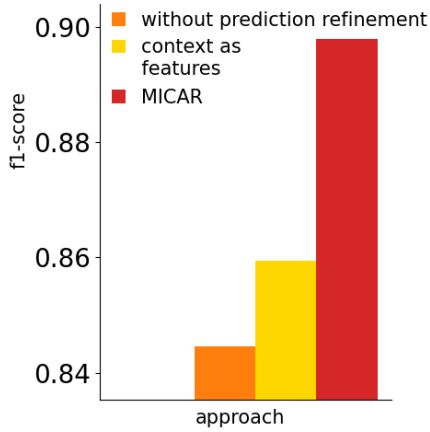


Figure 3.11: The impact of our prediction refinement approach on the recognition rate

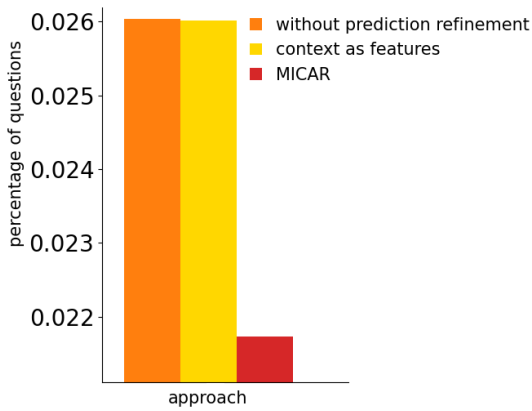


Figure 3.12: The impact of our prediction refinement approach on the percentage of questions

data significantly improve the recognition rate. Indeed, the *without context* approach reaches the lowest F1 score. Moreover, MICAR outperforms the *context as features* solution (+4%). This is due to the fact that ADLs can be performed in many different context situations. Considering high-level semantic data as fea-

tures makes the learning task more complex, thus requiring more labeled data. Moreover, our symbolic approach is more flexible since new context information can be added dynamically to the ontology, while the machine learning classifier should be re-trained from scratch if new features need to be considered.

Considering active learning queries, MICAR outperforms both approaches. Indeed, by discarding the inconsistent activities thanks to the PREDICTION REFINEMENT module, MICAR often increases its confidence in the remaining activities, thus reducing the percentage of triggered questions.

Predictions aggregation

Finally, we quantitatively evaluate the effectiveness of the PREDICTIONS AGGREGATION module in detecting jointly performed activities. For the sake of this evaluation, we only considered data from 2− and 4−subject scenarios.

We used the method proposed in Section 3.4.6 to compute group activities both on the classification output as well as on the ground truth. Figure 3.13 shows a confusion matrix that reveals MICAR's accuracy in detecting the correct number of users that are jointly performing an activity. Let $\langle A, U, [t_i, t_j] \rangle$ be a detected group activity where A is the joint activity, U is the set of users jointly performing A, and $[t_i, t_j]$ is the time interval of the group activity. Similarly, let $\langle A^\star, U^\star, [t_l, t_k] \rangle$ be a ground truth group activity where $A^\star$ is the joint activity, $U^\star$ is the set of users performing $A^\star$, and $[t_l, t_k]$ is the time interval of the ground truth group activity. We compare a predicted group activity and a ground truth group activity when $A = A^*$ and $[t_i, t_j] \cap [t_l, t_k] \neq \emptyset$ (i.e., the activity is the same and they temporally overlap). Hence, in this evaluation, we do not consider misclassifications, that are already captured by the results reported in Figure 3.6.

We consider a true positive when $U = U^\star$ (i.e., the set of users is exactly the same). We consider a false positive when $U \supset U^\star$ (i.e., the predicted group activity involves a higher number of users w.r.t. the ground truth). Finally, a false negative occurs when $(U \cap U^\star) \subset U^\star$ (i.e., only a subset of the users in the ground truth is actually in the prediction). From the confusion matrix, we observed that individual activities are sometimes detected as 2-subject activities. This is probably due to mistakes in data association. For instance, if Alice is

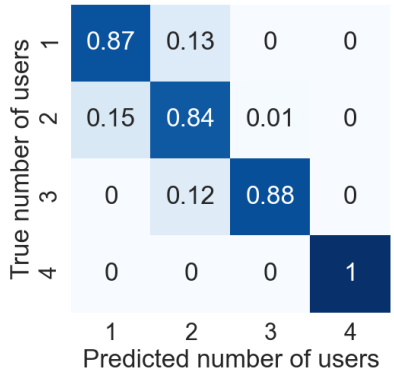


Figure 3.13: Confusion matrix on the number of users attributed to group activities

preparing a salad in the kitchen while Bob is cooking pasta in the same room, the electric cooker event could be mistakenly assigned to both users. Hence, MICAR could detect that Alice and Bob are cooking together over a certain interval of time.

Group activities are sometimes detected with a lower number of users compared to the ground truth. This could happen when MICAR performs a misclassification for a subset of users in the group. For example, suppose that Alice washed the dishes from $t0$ to $t3$, while Bob was clearing the table in the same time interval. From $t4$ to $t6$ they watched television together. MICAR may correctly predict Bob's activity while it may mistakenly detect that Alice washed the dishes from $t0$ to $t4$ and that she started to watch the television with Bob at $t5$. In this case, we count a true positive (Alice and Bob watched the television together from $t5$ to $t6$), but also a false negative (Bob individually watched the television from $t4$ to $t5$).

The accurate recognition rate for 4-subject activities is due to the fact that, in our dataset, only ADLs that are easy to detect (like eating and watching TV) are performed in this setting.

3.6 Discussion

3.6.1 Acceptability and privacy issues

MICAR is an ADL recognition system that continuously records the behavior of the users in their daily lives. Considering the application of our framework for healthcare applications, it may be perceived as a component of a therapy, hence it is more likely accepted with respect to other solutions. Moreover, MICAR does not consider intrusive devices like microphones and cameras. However, the data collected by MICAR are sensitive, and privacy measures should be considered in order to manage them. In our vision, the MICAR algorithms should run on a smart-home gateway and detailed sensor data should not be accessible from outside. In order to release information to healthcare stakeholders (e.g., clinicians), there are several solutions. Among them, aggregated ADLs data can be outsourced in an encrypted form to a cloud server. By relying on searchable encryption, it is possible to outsource encrypted data and, at the same time, to allow clinicians to perform queries on encrypted data [150].

Active learning may also be considered invasive and ethically inappropriate. Indeed, each query is an interruption to the daily life of a resident. Hence, active learning queries may not be acceptable if too frequent or if they are prompted at inappropriate times. While we show that the number of queries generated by MICAR is low and their frequency decreases quickly, in future work we will investigate a context-aware strategy in charge of prompting active learning queries based on the user's context, interrupting her only when appropriate.

3.6.2 Personalization

Personalization is an important aspect for accurate ADLs recognition [151]. This is also true for data association. Indeed, we believe that the additional personal high-level semantic information of the users may further improve data association. For instance, each user may have specific habits and routines, also depending on the role in the home (e.g., caregiver, elderly woman, elderly man). For instance, if the caregiver and an elderly subject are at the same time in the kitchen while the stove is being turned on, the caregiver is more likely the one triggering this event.

This high-level information can be considered as the system's prior knowledge or, alternatively, it can be automatically derived using pattern mining approaches that learn typical routines of each subject.

We also believe that the personal agenda of each user may help in providing hints about data association (e.g., if Alice has a dentist appointment in 20 minutes, she is more likely the one who is opening the door to leave home). In future work, we will investigate how to improve personalization aspects in MICAR.

3.6.3 Need for real-world experiments

A limitation of this work is that experiments are conducted using a public dataset acquired in a controlled setting. Experiments using real-world datasets are needed to better assess the effectiveness of our approach. We plan to perform this evaluation in the future, in the context of research projects related to healthcare.

Moreover, for the sake of this work, we did not consider the remote control of smart devices. Considering the specific sensors that we adopted in our experimental setup, only the smart plugs (controlling the TV and the stove) could actually be remotely controlled. Indeed, other devices like magnetic and mat sensors require physical interaction with the subject.

The remote control of smart devices introduces new challenges as we illustrate in the following example.

Example 3.4 *Alice is in the kitchen, while Bob is in the living room. Since Bob intends to prepare some food in the next few minutes, he decides to remotely turn on the oven to warm it up while he is still in the living room. MICAR mistakenly associates the event Turning ON oven to Alice, since she is the one actually in the kitchen.*

We believe that the data association strategy of MICAR can be extended to consider the remote activation of smart devices. For instance, when a resident controls a device by using her personal smartphone, the association is straightforward (i.e., the smartphone directly identifies the resident). However, smart devices may be also controlled using voice-based home assistants. In this scenario, a possible solution is to identify the users through the voice captured by the microphone.

Note that, considering Example 3.4, when Bob is turning on the oven from the living room, it is not clear if this event should be actually associated with him. Indeed, it is not trivial to determine what ADL classes the system should recognize when events related to remote control of smart devices are detected. We will investigate this direction in future work.

3.7 Summary

In this chapter, we presented MICAR, a novel framework for multi-subject HAR. This approach addresses the research question **Q1** presented in Section 2.5 by relying on symbolic reasoning to perform data association without requiring supplementary labeled data, thus mitigating labeled data scarcity. To the best of our knowledge, MICAR is the first work that combines Neuro-symbolic AI with semi-supervised learning to mitigate this issue for the recognition of ADLs in multi-subject smart environments. Our results showed how the symbolic data association strategy of MICAR allows the system to achieve results comparable with the ones of an ideal approach that performs data association based on ground truth. Moreover, MICAR reaches similar recognition rates compared to a fully supervised approach, while requiring significantly lower labeled data and triggering a limited number of active learning queries. One of the limitations of MICAR is that the PREDICTION REFINEMENT module refines the activity classifier's predictions by relying on rigid ontological reasoning procedures that cannot capture the intrinsic uncertainty of sensor data. For instance, consider a user who is washing the dishes in the kitchen. MICAR could derive a wrong user's posture (e.g., sitting) from the inertial sensor data collected by her smartwatch. This information is rigidly used to refine the prediction of the activity classifier, thus incorrectly discarding activities (like washing the dishes) that cannot be performed while sitting according to the proposed ontology. In the next two chapters of this thesis, we will investigate other less rigid Neuro-symbolic approaches for HAR that are based on the Knowledge Infusion paradigm. In particular, these methods will be investigated in another HAR domain affected by labeled data scarcity, i.e., the context-aware recognition of low-level activities on mobile/wearable devices.

Chapter 4

Knowledge infusion through symbolic features for context-aware HAR

4.1 Introduction

In the previous chapter, we have seen how Neuro-Symbolic AI (NeSy) solutions can be considered, even in combination with semi-supervised learning, to mitigate labeled data scarcity in multi-subject HAR. However, data scarcity also affects other application domains, such as the context-aware recognition of low-level activities (e.g., walking, running) through mobile and wearable devices. In this scenario, researchers introduced the use of contextual information about the users' surroundings like their semantic location and speed, or current local weather conditions [22]. This information has the potential to better discriminate activities with similar motion patterns, but executed in different context scenarios (e.g., sitting and sitting on transport). Unfortunately, it is not feasible to acquire comprehensive datasets that include every possible context condition in which activities may be performed by users.

Even in this case, NeSy methods can reduce the amounts of labeled data required to reliably build a context-aware activity classifier. However, like the PREDICTION REFINEMENT module of MICAR presented in Section 3.4.5, existing

NeSy solutions for context-aware HAR [1, 23] only consider domain knowledge to discard from the probability distribution generated by the activity classifier those activities that are not consistent with the user's surrounding context. This could lead to wrong decisions if the knowledge model is incomplete or in the presence of temporary noisy contextual information. For instance, GPS readings collected from the user's smartphone can be momentarily noisy, thus leading to consider incorrect contextual information about the user. Hence, in these situations, existing NeSy methods are too rigid and they could improperly discard wrong activities.

Knowledge Infusion is an emerging NeSy paradigm that may mitigate this problem since it aims to infuse domain knowledge directly into DL classifiers. In this way, the model internally learns domain constraints, while handling data uncertainty thanks to its data-driven learning process. In this chapter, we present a novel Knowledge Infusion method we designed for context-aware HAR. The features automatically extracted by a DL-based activity classifier from raw sensor data and high-level context data about the user's surroundings are combined with the ones inferred through symbolic reasoning. The symbolic features encode domain knowledge about the activities that are consistent with the surrounding context of the user and they are infused within the DL model, before its classification layer.

In particular, we will present two versions of our method. In the first version, symbolic reasoning relies on a standard ontology encoding hard constraints between context data and activities. For instance, this ontology may represent *running* as an activity implying that the user's current speed is positive. In the second version, we consider a probabilistic ontology composed of both hard and soft constraints, i.e., rules associated with a weight. For instance, in this ontology, the soft constraint *running can be performed indoors* has a lower weight than the soft constraint *running can be performed outdoors*.

Our results on two publicly available datasets for context-aware HAR indicate how the use of symbolic features mitigates data scarcity while being more robust than existing NeSy approaches in the presence of noisy context data. Moreover, we show how the improvements led by probabilistic ontologies do not justify the

significant effort required to build them.

The rest of the chapter is organized as follows. Section 4.2 formalizes context-aware HAR and formulates the NeSy context-aware HAR problem. Section 4.3 introduces our novel knowledge infusion method based on symbolic features. Section 4.4 presents the standard and the probabilistic ontologies we used as knowledge models, as well as their knowledge-based reasoning engines. Finally, Section 4.5 describes the experimental evaluation and the results obtained on two publicly available datasets for context-aware HAR, while Section 4.6 discusses the main limitations of the proposed method.

4.2 Preliminaries

In this section, we formalize context-aware HAR and we formulate the NeSy Context-Aware HAR problem. Moreover, we take advantage of this formalization to re-formulate existing NeSy strategies for Context-Aware HAR.

4.2.1 Context-Aware Human Activity Recognition

Let D_u be the dataset of raw sensor data collected from the mobile devices (e.g., smartphone, smartwatch) of a user u. Given a set of users $U = \{u_1, \ldots, u_n\}$, let $D^\star = \{D_{u_1}, \ldots, D_{u_n}\}$ be the set of datasets of all the users. Let $A = \{a_1, \ldots, a_k\}$ be the set of considered activities. The dataset $D^\star$ is associated with a set of annotations L that describes the activities performed by each user u. Each annotation $\lambda \in L$ is a tuple $\lambda = \langle u, a, t_s, t_e \rangle$ where a is a label identifying the activity actually performed by u during the time interval $[t_s, t_e]$. Each user dataset D_u is partitioned in a set of non-overlapping fixed-length windows $W_u = \{w_1, \ldots, w_q\}$ with each window including z seconds of consecutive raw sensor data of D_u.

In this work, we use the notion of *context* as a specific high-level situation that occurs in the environment surrounding and including the user while sensor data are being acquired (e.g., *it is raining*, *location is a park*, *current speed is high*). Let $C = \langle C_1, \ldots, C_p \rangle$ be a set of possible contexts that are meaningful for the application domain.

For each window w of raw data we identify two subsets w^R and w^C. The subset w^C includes raw sensor data that we consider useful to derive high-level contexts in C through reasoning and/or abstraction, while w^R includes raw data that we consider appropriate to be directly processed by a data-driven model (e.g., data from inertial sensors). Note that these subsets can have a non-empty intersection and their union is the whole w. The composition of w^R and w^C strictly depends on the target application, the available data, the knowledge model, and the available external services to obtain high-level context information.

Considering, for example, location data, it may be appropriate to exclude raw GPS coordinates from w^R and use it to obtain semantic location or other higher-level location information that can be more easily correlated with activities. On the other hand, leaving raw GPS data in w^R may not lead to a better model (it may be difficult to find correlations with activities and even when found, it may be difficult for the model to generalize).

Given w^C, let $ca(w^C)$ be a function named CONTEXT AGGREGATOR that derives all the contexts $C^w \subset C$ that are true during w based on w^C. This function can rely on simple rules, available services, or context-aware middlewares [14]. For instance, the geographical coordinates provided by the location service of the user's smartphone can be used to derive her semantic location (e.g., at home, in a public park) by querying a dedicated web service.

Definition 1 (Context-aware HAR) *Given a dataset $D^\star$ and the annotations set L, the problem of context-aware Human Activity Recognition is to provide to an unseen tuple $\langle w^R, C^w \rangle$, derived from a sensor data window w from user u, the probability distribution $P = \langle p_1, \ldots, p_k \rangle$, where p_i is the probability that u performed the activity a_i in contexts C^w, with $\sum_{i=1}^{k} p_i = 1$.*

4.2.2 Neuro-Symbolic Context-aware HAR

The *context-aware HAR* problem could be tackled by using purely data-driven models where context data are simply used as input. However, a more effective approach combines data-driven models with a knowledge model K that, based on a set of contexts C, encodes relationships between the activities in A and the contexts in C. For instance, according to common-sense knowledge, the activity

cooking is usually performed in a kitchen or, anyway, in a room equipped with a cooker, microwave, or oven. This relationship between the activity and the typical environment in which it is performed can be used in the HAR process, thus reducing the amount of labeled data required to learn it.

Note that K can be built in several different ways: by domain experts using common-sense knowledge on HAR, re-using existing knowledge bases (e.g., ontologies), or considering semi-automatic approaches in charge of extracting knowledge from external sources (e.g., text, images, and videos from the web). In any case, building a comprehensive and robust knowledge model is a challenging task. Even the knowledge of a domain expert is limited and is not guaranteed to capture all the possible context situations in which an activity can be performed [152].

Even though knowledge models cannot capture all the possible scenarios, our experiments will show their advantages in mitigating data scarcity when properly combined with data-driven methods. Indeed, in addition to the available training data, common-sense knowledge has the potential to capture constraints/patterns that are not learnable because of insufficient data. While there may be cases in which some rigid constraints would wrongly indicate the inconsistency between a context and an activity due to incompleteness, the knowledge model is supposed to model most of the usual context situations, and it can be refined and extended. Hence, we expect these cases to be rare. Also note that knowledge representation frameworks, like ontologies, have an open-world assumption. Hence, if reasoning cannot find an explicit inconsistency between a given context and an activity, their relationship is considered consistent.

Formally, given a knowledge model K and a set of contexts C^w, let $SR(K, C^w)$ be a function named SYMBOLIC REASONER that outputs, for each activity a_i, a likelihood value $l(a_i)$ (a value between 0 and 1) of a_i being consistent with the observed context C^w according to the constraints in K. Note that the majority of symbolic representation and reasoning approaches, including most ontologies, are based on formal logics that do not support uncertainty. In these cases $SR()$ will associate the value 1 to each a_i that is consistent with the observed context C^w according to the constraints in K, and the value 0 otherwise.

Definition 2 (Neuro-Symbolic Context-Aware HAR model) *A Neuro-Symbolic Context-Aware Human Activity Recognition model combines a deep learning model DNN and the symbolic reasoner function $SR()$ to solve the context-aware HAR problem.*

This very general definition is intended to capture in a single category approaches that combine in different ways the DNN and the $SR()$ modules as we will describe in Sections 4.2.3 and 4.3. Figure 4.1 graphically illustrates the high-level architecture of NeSy Context-Aware HAR shared by these approaches.

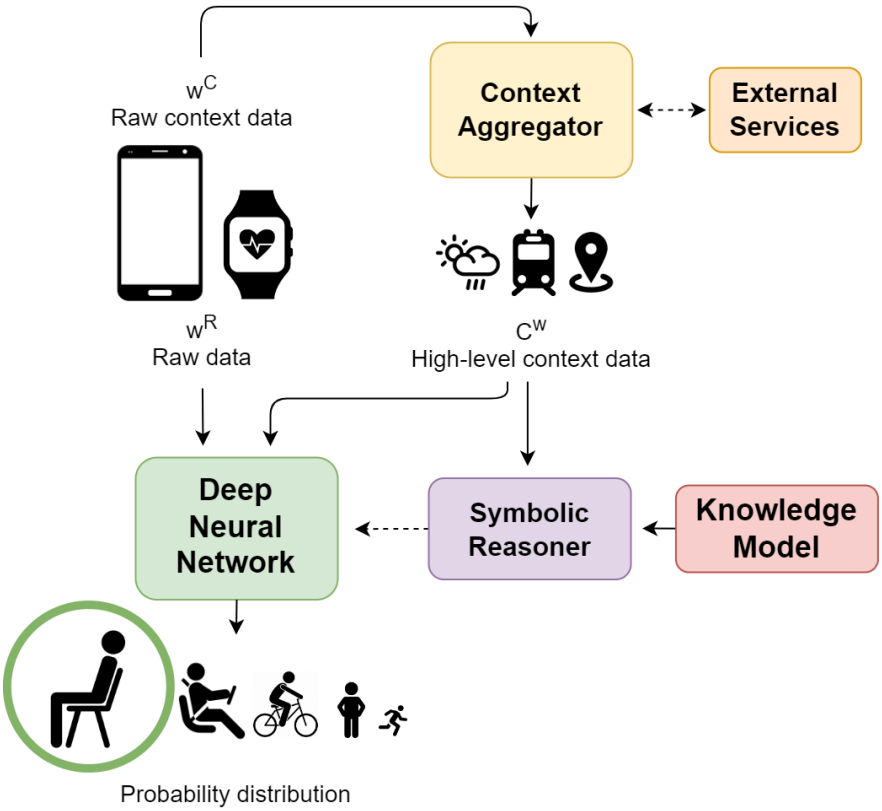


Figure 4.1: The neuro-symbolic context-aware HAR approach

4.2.3 Formalization of existing Neuro-Symbolic approaches

In this section, we re-formulate existing Neuro-Symbolic AI (NeSy) approaches with the notation introduced in sections 4.2.1 and 4.2.2 to compare them with our novel NeSy approach in an appropriate way. In particular, we consider the state-of-the-art approach for NeSy HAR, which is named *context refinement* [1]. Note that *context refinement* is the approach that inspired the PREDICTION REFINEMENT module of MICAR presented in Chapter 3.

The goal of the *context refinement* method is to a posteriori review the DNN predictions using the HAR knowledge encoded in K. As shown in Figure 4.2, the DNN is trained with the cross-entropy loss function $\mathcal{L}_{cross}$, which penalizes misclassifications on the training data. During classification, the output of the $SR()$ function is used to refine the probability distribution derived by DNN on a specific input. Intuitively, the likelihood values obtained by $SR()$ are used to reduce the probability of those activities that are less likely to be the correct predictions considering the current user's context.

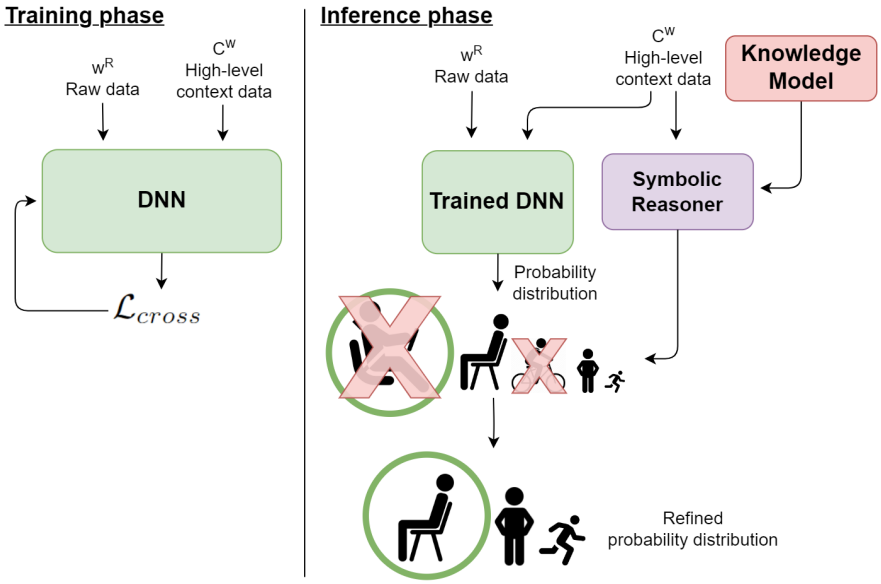


Figure 4.2: The *context refinement* neuro-symbolic approach [1]. In this example, two activities are excluded from the probability distribution since their likelihood, according to the Symbolic Reasoner module, is 0.

More formally, given a probability distribution $P = \langle p_1, \ldots, p_k \rangle$ emitted by

DNN on a tuple $\langle w^R, C^w \rangle$, and the likelihoods values provided by $SR(K, C^w)$, for each candidate activity a_i with $i = 1, \ldots, k$ we compute $p_i * l(a_i)$ and then normalize in order to obtain a *knowledge-refined probability distribution*.

Note that when symbolic reasoning is based on a standard ontology, $l(a_i)$ is a binary value and the above operation is equivalent to excluding some of the activities from the candidates and normalizing.

The objective of *context refinement* is to correct wrong decisions made by DNN, thus increasing its recognition rate. At the same time, it ensures that each classified activity is consistent with the surrounding context of the user. A drawback of this approach is that most ontology-based reasoning may encode rigid constraints about the relationships between contexts and activities, resulting in *context refinement* discarding activities that are occasionally performed in unusual context scenarios (e.g., the knowledge engineer may explicitly exclude that the activity *running* can be performed at *the mall*, as a semantic place).

In the following, we report a simplified running example of the *context-refinement* approach:

Example 4.1 *Consider an activity classifier trained offline in a supervised fashion by a service provider using a labeled dataset. After training, the classifier and a symbolic reasoner based on a standard ontology are deployed on Alice's smartphone to recognize her activities in real time. Suppose that Alice is sitting, and the smartphone collects a window* $\langle w^R, C^w \rangle$ *of raw sensor data and high-level context data during the execution of this activity. Given this window, the classifier outputs the following probability distribution: Walking:* 50%, *Sitting:* 30%, *Standing:* 15%, *Running:* 5%. *We observe that the most likely activity is walking, which is not correct according to the ground truth. The high-level context* C^w *encodes the information that Alice's current speed is* 0. *By processing* C^w, *the symbolic reasoner infers that the likelihood of Walking and Running is* 0 *(since they can not be performed with null speed), while the likelihood of the other activities is* 1. *By multiplying each probability value with the corresponding likelihood and normalizing the resulting values, a new probability distribution is obtained: Sitting:* 67%, *Standing:* 33%, *Walking:* 0%, *Running:* 0%. *After refining the probability distribution, the most likely activity is sitting which corresponds with*

the actual activity performed by Alice.

4.3 Knowledge infusion through symbolic features

The concept of introducing a knowledge infusion layer in a DNN was originally proposed in [53]. The objective of the *symbolic features* is to directly incorporate the knowledge encoded in K into DNN, not only at the inference phase but also during the learning process. Hence, the *symbolic features* method allows the DNN also to learn the correlations between input data and context-consistent activities. Compared to *context refinement*, this approach is more robust to noisy input data or to an incomplete knowledge model since domain constraints are directly learned by the DL model, while data uncertainty is handled thanks to its data-driven learning process.

As depicted in Figure 4.3, the information about the context-consistency of

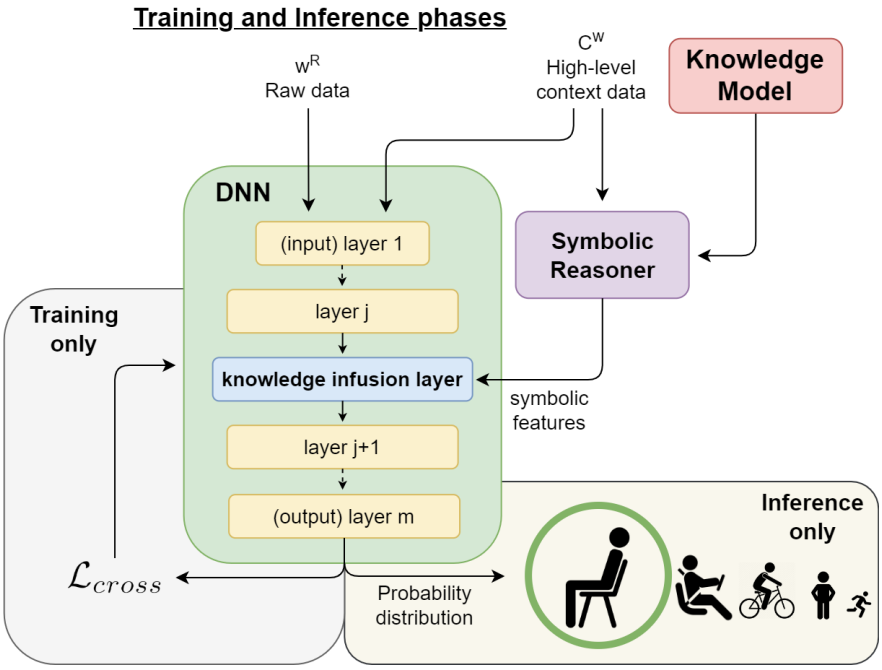


Figure 4.3: The *symbolic features* neuro-symbolic approach

activities provided by $SR()$ is used to generate symbolic features that are infused

within the hidden layers of DNN through a dedicated layer named *knowledge infusion* layer. More formally, given an input tuple $\langle w^R, C^w \rangle$, and the likelihood values provided by $SR(K, C^w)$, the symbolic features consist of a vector f_s in which the i-th element is $l(a_i)$. Similarly to context refinement, please note that if symbolic reasoning is not probabilistic f_s is a binary vector. Section 4.4 will present the two alternative ontologies (i.e., a standard and a probabilistic) that we considered to realize the SYMBOLIC REASONER module.

Given the sequence of DNN's layers $\ell_1, \ldots, \ell_m$, and the symbolic features f_s generated through $SR()$, the *symbolic features* method adds to DNN a *knowledge infusion* layer ℓ_{ki}. This layer receives as input the symbolic features f_s and the features automatically extracted by a DNN's hidden layer ℓ_j with $1 < j < m$. Then ℓ_{ki} concatenates in the latent space the features received as input and generates a novel feature vector that is provided to the next layer ℓ_{j+1}. Also in this case, the DNN is trained through the cross-entropy loss function $\mathcal{L}_{cross}$.

This methodology is less rigid than *context refinement* in excluding some activities based on knowledge consistency since domain knowledge is infused into the data-driven model instead of just being used afterward, to modify the result of the neural network.

In the following, we report a simplified running example of the *symbolic features* approach:

Example 4.2 *A service provider trains, in a supervised way, an activity classifier using a labeled dataset and a symbolic reasoner based on a standard ontology. For each window $\langle w^R, C^w \rangle$, the symbolic reasoner analyzes C^w to obtain the likelihood values for each activity, that are used to generate symbolic features. For instance, when C^w includes home as semantic location, the symbolic feature corresponding to the driving activity is* 0*. The model is trained by providing windows of raw sensor data and high-level context data in the input layer, while symbolic features are given to the knowledge infusion layer. After training, the classifier and the reasoner are deployed on Alice's smartphone to recognize her activities in real time. Suppose that Alice is sitting, and the smartphone collects a window $\langle w^R, C^w \rangle$ of raw sensor data and high-level context data during the execution of this activity. The high-level context C^w encodes the information that Alice's cur-*

rent speed is 0. *By processing* C^w, *the symbolic reasoner generates a symbolic feature vector, where Walking and Running have value* 0 *(since they can not be performed with null speed), while the remaining activities have value* 1. *In order to perform classification, the window* $\langle w^R, C^w \rangle$ *is provided to the input layer, and the symbolic feature vector is provided to the knowledge infusion layer. Thanks to the information encoded in the symbolic features, the classifier will assign a lower probability value to Walking and Running, since it has learned during training that these activities are inconsistent according to the symbolic features.*

4.4 Ontological models

The SYMBOLIC REASONER is in charge of inferring the symbolic features that will be infused within the DL classifier during both training and inference. To achieve this goal, this module relies on a knowledge model (i.e., an ontology in our implementation) that encodes the relationships between context information and activities.

Ontologies are currently the most widely used formalism to represent and reason about common knowledge and context data [45]. Compared to simple rules, the ontology representation that we adopt has the advantage of enabling hierarchical and relational reasoning; for example, the relationship between a location context (e.g., a public park) and an activity class (e.g., static physical activities) is inherited by more specialized activities in a subclass (e.g., sitting, standing). This means that the ontology captures implicit rules and enables reasoning based on rule chaining. Ontologies adopt an open-world assumption, hence if a relationship or fact cannot be derived as false it may be true. However, note that some strict constraints can be formulated, for example stating that the activity *sitting on transport* can only take place while the user is following a public transportation route. Hence, if location context data reveals that the user is not following one of these routes, that activity is considered inconsistent.

Periodically, high-level context data are automatically translated into ontological facts, which are then added to the ontology as a description of the current surrounding context of the user. Hence, the SYMBOLIC REASONER uses the ontology to infer, for each activity, a likelihood value about its consistency with respect

to the observed context. In particular, in this thesis, we considered two different ontologies: a standard ontology described in Section 4.4.2, and a probabilistic ontology presented in Section 4.4.3.

4.4.1 Translating context data into ontological facts

The high-level context data provided by the CONTEXT AGGREGATOR are automatically mapped to ontological concepts by a specifically designed middleware. This encodes the necessary rules to transform high-level context data into high-level axioms. Most of the context data we considered have a one-to-one mapping with ontological entities. For instance, the user's semantic location obtained from public web services is automatically mapped to the corresponding ontological fact.

On the other hand, raw context data available as scalar values are discretized by the CONTEXT AGGREGATOR. For instance, each user's speed value is mapped to one of the following ontological concepts: NullSpeed, LowSpeed, MediumSpeed, and HighSpeed. The rules used to discretize scalar values rely on ranges of values designed by knowledge engineers (e.g., speed values greater than 0 km/h and lower than 4 km/h are mapped to LowSpeed).

4.4.2 Standard ontology

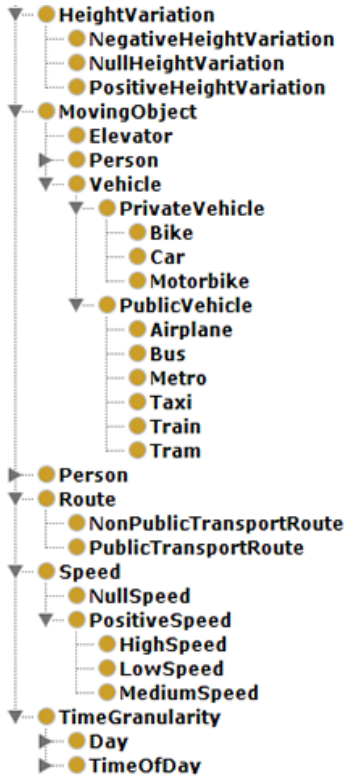
The *standard ontology* we considered in this thesis is an extension of the one proposed in the paper where *context refinement* was introduced [1]. We took advantage of the Protégé tool[1] to extend this ontology to better cover the taxonomy of activities and their relationship with context data for the datasets that we will describe in Section 4.5.1.

Standard ontology modeling

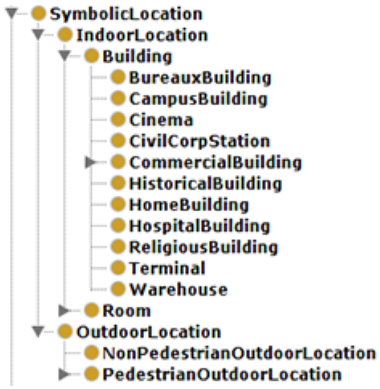
Our standard ontology considers several sources of context data: user's semantic place, user's presence in an indoor or outdoor setting, user's speed, user's proximity to public transportation stops and routes, user's height variations, local weather conditions, and temporal context (e.g., time of the day and day of the

[1]https://protege.stanford.edu/

week). Figure 4.4a shows a portion of the context information modeled in our standard ontology, while Figure 4.4b focuses on the set of considered semantic locations.



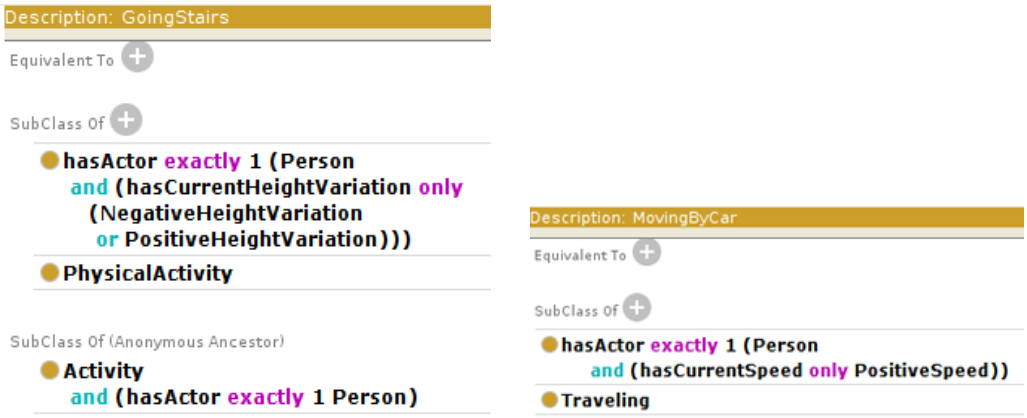
(a) An excerpt of the context hierarchy of our standard ontology



(b) An excerpt of the symbolic locations hierarchy of our standard ontology

Figure 4.4: Excerpts of our standard ontology

Due to the intrinsic open-world assumption of ontologies, we explicitly state the necessary conditions that make activities possible or not possible in a given context. As we will explain later, such constraints are necessary to enable the generation of symbolic features that are based on *consistency* reasoning. For instance, as shown in Figure 4.5a, the activity *taking stairs* (or *going stairs*) should take place when the user experiences a positive or negative height variation. Another example is the activity *moving by car* (Figure 4.5b): our standard ontology enforces that it should take place when the user's speed is positive.



(a) Definition of the activity *taking stairs* (or *going stairs)*

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

Figure 4.5: Examples of activity definitions in our ontology

Standard symbolic reasoning

Considering a standard ontology, we use the ontology *consistency checking* as the symbolic reasoner function $SR()$ defined in our formalization. In particular, for each activity, we evaluate if it is consistent considering the available context data. For instance, the activity *running* is consistent only when the user is experiencing a positive speed. Context-consistent activities are associated with 1 as likelihood, while context-inconsistent activities are associated with 0.

To check whether an activity a_i is context-consistent, our method adds to the terminological part of the ontology an axiom representing an instance of `Person` which identifies the user. Then, available context data are represented as ontological concepts, as explained in Section 4.4.1. Hence, we add an axiom stating that the user is performing the activity a_i. Finally, we rely on ontological reasoning (using the Pellet reasoner [153]) to check if a_i is consistent with the user's context.

Example 4.3 *Alice is using our system based on a standard ontology. When the ontological reasoning task is triggered,* `Person(Alice)` *is added as a fact. Then, context data are analyzed to expand the set of facts. Suppose that by interacting with a web service, the* CONTEXT AGGREGATOR *module derives that Alice is in a park and that her current speed as obtained by the GPS of her smartphone is 10*

km/h. This context information is used to automatically instantiate the following individuals in the ontology: `Park(place)` *and* `MediumSpeed(speed)`*. Then, the following relationships between Alice and context data are added as facts:* `hasCurrentSymbolicLocation(Alice, place)` *and* `hasCurrentSpeed(Alice, speed)`*. Finally, to check if the activity* ***running*** *is context-consistent, our system adds the following axioms:* `Running(currentActivity)` *and* `isPerforming(Alice, currentActivity)`*. The consistency of this set of facts with the domain constraints encoded in the standard ontology will determine if* ***running*** *is consistent according to the current surrounding context of Alice.*

4.4.3 Probabilistic ontology

Since our definition of symbolic reasoning on the knowledge model admits also fuzzy or probabilistic methods, in this thesis, we also consider a *probabilistic ontology* based on log-linear description logics [154]. In particular, we slightly extended the knowledge model proposed in [64], which is an extension of the model originally proposed in [1].

Probabilistic ontology modeling

Our probabilistic ontology combines *hard* constraints (i.e., rigid rules that are always true) and *soft* constraints (i.e., rules associated with weights) to model relationships between contexts and activities. Hard constraints capture context conditions that should always be satisfied to consider a given activity as possible. An example of a hard constraint is *running implies a user positive speed*. On the other hand, soft constraints capture context conditions that can occur when an activity is performed, but they are not required to be always verified. This behavior can be obtained by associating a certain degree of confidence with the ontological axiom. For instance, the soft constraint *running can be performed indoors* has a lower weight than the soft constraint *running can be performed outdoors*. Intuitively, the weight associated with a soft constraint expresses a degree of compatibility between an activity and a specific context information.

Axioms' weights In log-linear description logics, the weight associated with a soft axiom takes values in $\mathbb{R}$. We associated with each axiom a probability weight $pw \in [0, 1]$ based on common-sense knowledge about HAR. For instance, we associated the weight 0.1 to the soft axiom *running can be performed indoors*, while 0.9 to the soft axiom *running can be performed outdoors*.

In order to approximate probability values for a log-linear model, as proposed in other works [155], we use the *logit* function to map each weight pw to a real number as follows:

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

Note that *logit* is not defined at 0 and at 1. When $pw = 1$ or $pw = 0$ 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 symbolic reasoning

Once the probabilistic ontology has been extended with facts about the current surrounding context of the user, we rely on the probabilistic reasoner ELOG [156] to derive the consistency likelihood $l(a_i)$ of each activity a_i according to the hard and soft constraints included in the ontology. To build the symbolic features vector infused into the DL-based activity classifier, we mapped $l(a_i)$ to $[0, 1]$ in order to be consistent with the formalization presented in Section 4.2.

4.5 Experimental evaluation

In this section, we describe the experimental evaluation that we carried out to assess the quality of our method based on symbolic features compared to a baseline and the state-of-the-art NeSy approach introduced in Section 4.2 (i.e., *context refinement*). First, we introduce the two datasets that we considered for the evaluation. Then we describe our experimental setup: how we pre-processed the datasets, the models used and the evaluation methodology adopted. Finally, we present the results of our evaluation.

4.5.1 Datasets

The evaluation of context-aware HAR approaches requires datasets including both inertial sensor data and contextual information. However, there are a few publicly available datasets with such characteristics. Existing NeSy approaches for context-aware HAR have been evaluated only on scripted and non-public datasets [1]. In this thesis, we consider a scripted dataset that we collected and published in a parallel work and a publicly available in-the-wild dataset, both including sensor and context data.

DOMINO

DOMINO [24] is a HAR dataset we collected and recently published as parallel research in our research lab. DOMINO includes several context-dependent activities monitored through mobile devices that collected both inertial sensor data and high-level context data. In particular, DOMINO includes data from 25 subjects wearing a smartwatch on their dominant hand's wrist and a smartphone in their pocket. Raw sensor data have been collected from the inertial sensors (accelerometer, gyroscope, and magnetometer) installed on both these mobile devices. At the same time, the dataset also includes high-level context data collected by combining public web services and the smartphone's built-in sensors. The measurements of the barometer and the GPS of the smartphone were discretized to provide information about the users' height and speed variations. Moreover, the dataset incorporates the output of the following web services: (1) Google's Places API provided the semantic places closest to the user; from this information, it was also derived the presence of the user in an indoor or an outdoor environment; (2) OpenWeatherMap provided current local weather conditions (e.g., sunny), while (3) Transitland provided transportation routes and stops close to the user; the combination of this information with location data was used to derive whether the user was following a public transportation route. DOMINO was acquired in a scripted fashion: the volunteers were asked to perform a sequence of indoor/outdoor activities, but they were not told how to execute them. Also, the volunteers were monitored by the research staff during data acquisition. As a consequence, the variability of context situations is limited. Overall, DOMINO

contains almost 9 hours of labeled data ($\approx$ 350 activities instances), including 14 different types of activities: *brushing teeth*, *cycling*, *elevator down*, *elevator up*, *lying*, *moving by car*, *running*, *sitting*, *sitting on transport*, *stairs down*, *stairs up*, *standing*, *standing on transport*, and *walking*.

ExtraSensory

ExtraSensory [15] is a public dataset for context and activity recognition. It includes inertial and context data collected in the wild from mobile devices of up to 60 users. Inertial data were collected through each user's personal smartphone (including both iOS and Android devices) and from a smartwatch provided by the researchers. More specifically, the dataset includes raw data measured by the accelerometer, the gyroscope, and the magnetometer of the smartphone, and raw data collected by the accelerometer of the smartwatch. Besides providing raw sensor data, ExtraSensory also provides data as handcrafted feature vectors (138 features) extracted from the raw measurements collected through inertial and other smartphone sensors (e.g., microphone, luminosity sensor) in 20-second time windows. Overall, ExtraSensory contains about $300k$ minutes of labeled data, including 51 different labels self-reported by the users and encoding both high-level context information (e.g., at home, with friends, phone in bag, phone is charging) and performed activities (e.g., sitting, bicycling). Since it has been collected in the wild, different research groups in the HAR community used ExtraSensory to assess the generalization capabilities of activity recognition frameworks in real-world scenarios [157, 158]. Due to the complexity of the dataset, existing HAR methods evaluated on ExtraSensory achieved low recognition rates. For instance, by considering as input the raw inertial measurements provided by the accelerometer and the gyroscope of the smartphones, the CNN-based method proposed in [157] reached an average macro F1 score of ≈ 0.53, only considering 4 target activity classes: *idle* (*lying* or *sitting*), *walking*, *running*, and *cycling*. In another work, by considering the handcrafted features of ExtraSensory, an AdaBoost classifier reaches ≈ 0.63 of average macro F1 score on 5 target activities (i.e., *walking*, *standing*, *sitting*, *exercise*, and *sleeping*) [158]. Hence, this dataset represents a challenging benchmark.

4.5.2 Experimental setup

In the following, we describe our experimental setup.

Data pre-processing

Consistently with existing works proposing NeSy approaches for Context-Aware HAR [1], for both datasets, we segmented sensor data into non-overlapping windows of $k = 4$ seconds. For each raw data window w, we considered in the subset w^R only the data from inertial sensors, while part of the rest of the data would be much more helpful in its aggregated high-level form (C^w).

In the following, we describe the specific pre-processing steps we adopted for each dataset.

DOMINO Considering *DOMINO*, we planned to recognize all the 14 different available activities, by considering the raw inertial measurements collected by the accelerometer and the gyroscope of the smartphone and the smartwatch. Moreover, in our experiments, we considered 6 different context information types: the presence of the user in *indoor/outdoor* locations, her *semantic place* (e.g., home, workplace, gym, bar), her discretized *speed* (i.e., null, low, medium, high), her *proximity to public transportation routes*, her discretized *height variation* (i.e., negative, null, positive), and the *weather conditions* (e.g., sunny, rainy). Table 4.1 shows the number of samples involved during our experiments for each activity class of *DOMINO*.

ExtraSensory Considering *ExtraSensory*, we planned to recognize 7 different activities: *bicycling*, *lying down*, *moving by car*, *on transport*, *sitting*, *standing*, and *walking*. Specifically, for the activity class *walking* we consider those samples labeled as *walking* and/or *strolling* in the original dataset. For *moving by car*, we consider samples labeled with *in a car*, *car driver*, and/or *car passenger*, even when coupled with the label *sitting*. Finally, we labeled as *on transport* those samples originally labeled with *sitting* or *standing* coupled with the label *on a bus*.

Table 4.1: Number of samples for each activity class in *DOMINO*

Activity	**Number of samples**
Brushing teeth	163
Cycling	323
Elevator down	171
Elevator up	110
Lying	387
Moving by car	188
Running	334
Sitting	1764
Sitting on transport	213
Stairs down	266
Stairs up	187
Standing	1875
Standing on transport	297
Walking	1378
Total	**7656**

Table 4.2: Number of samples for each activity class in *ExtraSensory*

Activity	**Number of samples**
Bicycling	2920
Lying down	3055
Moving by car	2150
On transport	610
Sitting	23905
Standing	14280
Walking	11230
Total	**58150**

Before conducting our experiments, we performed some steps of data cleaning. First of all, we considered only those samples including inertial measurements recorded from the accelerometer and the gyroscope of the smartphone and from the accelerometer of the smartwatch. Indeed, for some users of *ExtraSensory*, gyroscope data from smartphones are not available. Moreover, not all of the dataset's users wore the smartwatch during data collection. Then, based on the available self-reported labels, we discarded the data collected while the smartphone's user was in a bag, or on a table. Indeed, we considered only phone positions that have been commonly considered in the literature (i.e., in the pocket and in hand). Finally, since the labels of *ExtraSensory* were self-reported by the users involved in the data collection, we discarded samples that we considered unreliable, due to the fact that they included self-reported labels not consistent with the recorded data. For instance, we discard segmentation windows including positive speed values but labeled with static physical activities like *lying*. As another example, we discarded those samples simultaneously labeled with *in a car* and *at home*. Table 4.2 shows the number of samples for each activity class of *ExtraSensory* after data cleaning. Note that, after our data cleaning process, we considered data overall from 31 subjects.

As inertial sensor data, we considered the raw data measured from the accelerometer and the gyroscope of the smartphone and from the accelerometer of the smartwatch.

Regarding context data, we considered the ones that can be easily derived from sensors of mobile/wearable devices. For instance, we considered the information about the user's semantic place (e.g., at the beach) since it could be derived by combining localization data and external web services, but not the position of the user's smartphone (i.e., in the pocket, in hand). In some cases, we discretized available information: for instance, the *speed* values observed thanks to the GPS were discretized into *null/low/medium/high speed*. Other high-level context information was obtained by directly considering available data, like *audio level*, *light level*, *screen brightness*, *battery plugged AC/USB*, *battery charging*, *on the phone*, *ringer mode normal/silent/vibrate*, and the time of the day (e.g., *Time 0-6*, *Time 18-24*). Moreover, we relied on the self-reported label *on a bus*, assuming that similar information could be derived by combining GPS data and web services like *Transitland*, as we did in *DOMINO*. Finally, we considered the semantic locations self-reported by the subjects (i.e., *home*, *workplace*, *school*, *gym*, *restaurant*, *shopping*, *bar*, *beach*). As already mentioned, semantic location information can be derived, for instance, by combining location coordinates data with *Google's Places API*.

DNN's architecture

The DNN we used for our experiments receives as input three separate inputs for each segmentation window: a) the smartphone's inertial sensors data, b) the smartwatch's inertial sensors data, and c) the one-hot encoded high-level context data[2].

Similarly to existing works, we rely on convolutional neural networks to capture spatio-temporal dependencies of sensor data [159, 160, 161, 162]. Even though more sophisticated networks have been proposed in the literature, in this work we use a simple solution to focus on the contribution of knowledge. The exact structure of our own CNN model has been determined empirically. Specifically, inertial sensors' data from the smartphone are processed by three *convolutional layers* composed of 32, 64, and 96 filters with a kernel size equal to

[2]Note that, we did not include raw context data as input since it is intuitively easier to learn correlations between activities and high-level context (e.g., semantic place) rather than between activities and raw context (e.g., geographical coordinates).

24, 16, and 8, respectively. These layers are separated by *max pooling* layers with a pool size of 4. After the three *convolutional layers*, we add a *global max pooling* layer, followed by a *fully connected* layer that includes 128 neurons. The smartwatch inertial sensors' data are provided to another component of DNN that presents the same sequence of layers used to automatically extract features from the smartphone's inertial data. The only difference is that, in this case, the three *convolutional layers* present a kernel size of 16, 8, and 4, respectively. Finally, the high-level context data is provided to a single *fully connected* layer composed of 8 neurons. The features extracted by these three independent flows are then combined thanks to a *concatenation* layer, which is followed by a *dropout* layer with a dropout rate of 0.1, and a *fully connected* layer with 256 neurons, useful to extract meaningful correlations between the concatenated features. The last layer of the network is a *softmax* layer that is in charge of providing a probability distribution over the possible activities.

In our experiments, we use this DNN architecture in three different ways:

- As a purely data-driven *baseline*, without further modifications
- Enhanced by combining in the *concatenation layer* the *symbolic features* and the features automatically extracted from input data (see Section 4.3)
- As the *DNN* module of the *context refinement* approach (see Section 4.2.3)

Cross-validation

We evaluated the approaches presented in Sections 4.2.3 and 4.3 by adopting the *leave-k-users-out* cross-validation technique. At each fold, k users are used to populate the test set, while the remaining users are used to populate training (90%) and validation (10%) sets. We also simulated several data scarcity scenarios by downsampling the available training data at each fold (e.g., 1%, 50%).

Considering the *DOMINO* dataset, we considered $k = 1$ (leave-one-user-out cross-validation). On the other hand, as also done by other works in the literature [157], for the *ExtraSensory* dataset we choose $k = 5$. At each iteration, we used the test set to evaluate the recognition rate of the different approaches in terms of the F1 score.

For the sake of robustness, we run each experiment 5 times, computing the average F1 score and the 95% confidence interval. Overall, the training process was based on a maximum of 200 epochs and a batch size of 32 samples. We considered an *early stopping* strategy, stopping the learning process when the loss computed on the validation set did not improve for 5 consecutive epochs.

4.5.3 Results

In the following, we show how our *symbolic features* approach outperforms a purely data-driven classifier in terms of recognition rate both in scripted and in-the-wild scenarios. We also compare our method with the *context refinement* approach presented in Section 4.2.3. Although our method seems inferior to *context refinement* in data-scarce scenarios, it is a more robust neuro-symbolic solution in the presence of noisy context data.

Our main results consider the standard ontology presented in Section 4.4.2 as the knowledge model since it is a widely used knowledge and context representation framework. The results using the probabilistic ontology introduced in Section 4.4.3 are presented in Section 4.5.3.

Comparison with other approaches

Tables 4.3 and 4.4 compare our *symbolic features* method with: i) the purely data-driven *baseline*, and ii) the *context refinement* strategy. More specifically, we considered different percentages of available training data for each dataset, thus comparing the approaches in different data scarcity scenarios.

Overall, on each dataset, the NeSy approaches outperform the *baseline*, considering all the data scarcity scenarios. This result suggests that traditional symbolic AI approaches have the potential to enhance the predicting capabilities of purely data-driven deep learning models.

Focusing on the scripted scenarios of *DOMINO* (Table 4.3), when the availability of labeled data is drastically low, *symbolic features* is worse than *context refinement* ($\approx +13\%$ against $\approx +23\%$). These performance differences become progressively smaller while increasing training data availability. Indeed, from 30% of training data, *symbolic features* and *context refinement* reach similar results.

Table 4.3: DOMINO: Results in terms of macro F1 score and 95% confidence interval

Training set percentage	**Baseline**	**Symbolic features**	**Context refinement**
10%	0.5946 (±0.008)	0.7268 (±0.008)	**0.8192** (±0.009)
20%	0.7529 (±0.010)	0.8590 (±0.012)	**0.8811** (±0.007)
30%	0.8268 (±0.006)	**0.9107** (±0.011)	0.9078 (±0.009)
40%	0.8556 (±0.011)	0.9152 (±0.009)	**0.9178** (±0.005)
50%	0.8835 (±0.011)	0.9198 (±0.011)	**0.9281** (±0.012)
60%	0.8917 (±0.010)	0.9237 (±0.009)	**0.9305** (±0.006)
70%	0.8915 (±0.006)	**0.9265** (±0.004)	0.9225 (±0.004)
80%	0.9007 (±0.007)	0.9254 (±0.008)	**0.9274** (±0.005)
90%	0.8965 (±0.002)	**0.9277** (±0.007)	0.9232 (±0.002)
100%	0.9024	**0.9408**	0.9221

Table 4.4: ExtraSensory: Results in terms of macro F1 score and 95% confidence interval

Training set percentage	**Baseline**	**Symbolic features**	**Context refinement**
1%	0.3127 (±0.023)	0.3418 (±0.010)	**0.6324** (±0.014)
2.5%	0.4279 (±0.008)	0.4720 (±0.016)	**0.6540** (±0.003)
5%	0.4867 (±0.013)	0.5877 (±0.025)	**0.6797** (±0.003)
7.5%	0.5167 (±0.016)	0.6359 (±0.008)	**0.6656** (±0.004)
10%	0.5199 (±0.011)	0.6534 (±0.012)	**0.6622** (±0.007)
25%	0.5842 (±0.016)	0.6404 (±0.010)	**0.6483** (±0.010)
50%	0.6096 (±0.007)	0.6216 (±0.007)	**0.6258** (±0.007)
75%	0.5813 (±0.032)	**0.6268** (±0.007)	0.6067 (±0.023)
100%	0.6053	**0.6205**	0.6190

In particular, when all the available training data are considered (i.e., 100%), *symbolic features* outperforms *context refinement* by $\approx +2\%$.

Similar insights are observed when focusing on the realistic scenarios of *ExtraSensory* (Table 4.4). Here, from 7.5% of training data, *symbolic features* and *context refinement* reach similar recognition rates. Note that, due to the complexity of the dataset, we achieved relatively low recognition rates on *ExtraSensory* (e.g., the max F1 score is ≈ 0.68). As described in Section 4.5.1, our results are in line with other works on the same dataset [157, 158].

Although *context refinement* seems better than *symbolic features*, in the next section we will see how our method is a less rigid solution that is more robust in the presence of noisy context data. Moreover, as we will discuss in Section 5.4, knowledge infusion methods like *symbolic features* seem to increase the interpretability of deep learning activity classifiers.

Robustness to noise

In order to show that our *symbolic features* method is more robust to uncertainty than *context refinement* even considering a standard ontology, we performed another set of experiments by introducing noise in the test data. In particular, we performed different experiments considering 5%, 10%, and 15% of noisy data in the test set. More specifically, for each perturbed data sample, we modified the semantic location context with another one (plausibly not too distant from the real one) that the knowledge model considers inconsistent with the ground truth activity. This perturbation simulates noise in GPS data acquired from mobile devices, often impacting the actual semantic location where the user is located. For instance, a subject at home may be wrongly located at a coffee shop that is in a nearby building.

Table 4.5 shows the results of this experiment. We observe that noise has the most negative impact on *context refinement*, thus confirming that it is the most rigid approach. Indeed, by discarding activities that are not consistent with the current context, this approach is the one suffering more from noisy context data. On the other hand, *symbolic features* is a less rigid NeSy approach that is able to mitigate this issue. Finally, we observed that the *baseline* method is the

Table 4.5: Average results with 5 different runs in terms of macro F1 score, considering 10% of training data and different percentages of dirty samples in the test set

	Original test set	**5% of dirty test set (delta)**	**10% of dirty test set (delta)**	**15% of dirty test set (delta)**
Baseline	0.5199	0.5089 (- 0.0110)	0.4983 (- 0.0216)	0.4954 (- 0.0245)
Symbolic features	0.6534	**0.5498** (- 0.1036)	**0.5200** (- 0.1334)	**0.5043** (- 0.1491)
Context refinement	**0.6622**	0.5430 (- 0.1192)	0.5057 (- 0.1565)	0.4801 (- 0.1821)

approach most robust to uncertainty, due to better generalization capabilities. Nonetheless, our *symbolic features* method still outperforms the *baseline* in each considered setting, hence confirming the advantage of infusing knowledge in deep learning models.

Results with a probabilistic ontology

Table 4.6 summarizes the results that we obtained on both datasets by using a probabilistic knowledge model slightly adapted from the one proposed in [64]. For

Table 4.6: Average results with 5 different runs in terms of macro F1 score, considering a data scarcity scenario simulated by using 10% of training data and the probabilistic version of each method

	DOMINO		**ExtraSensory**	
	Standard	Probabilistic	Standard	Probabilistic
Baseline	0.5946	0.5946	0.5199	0.5199
Symbolic features	0.7268	**0.7365**	**0.6534**	0.6408
Context refinement	0.8192	**0.8399**	0.6622	**0.6793**

the sake of simplicity, we show the results considering the data scarcity scenario where only 10% of labeled data are available. Our results indicate that, in general, introducing fuzziness only slightly improves the recognition rate obtained by the approach based on a standard ontology. The maximum improvement is $\approx +2\%$ on the DOMINO dataset. The only case where the probabilistic approach is

slightly worse than the deterministic one is by using symbolic features on the ExtraSensory dataset. This is likely due to the fact that, on this dataset, it often happens that the ground truth activity is not always the one corresponding to the symbolic feature with the highest likelihood. This aspect significantly complicates the learning process since this method probably heavily relies on the infused symbolic features during classification, which leads the model to predict the activity with the highest likelihood in the symbolic features vector. On the other hand, considering the deterministic case, consistent activities are always associated with a symbolic feature with a value of 1, thus avoiding this problem.

We believe that the small improvement in the recognition rate does not justify the effort of designing and managing probabilistic ontologies. Indeed, such models require significant effort in deciding the weights associated with soft constraints, that should capture general aspects of activities execution. Hence, we believe that relying on standard ontologies to capture the most common situations is an appropriate choice when coupled with knowledge infusion methods since they reduce the modeling effort while maintaining good recognition rates.

4.6 Discussion

4.6.1 Context data collection

In this work, we assume that context data can be continuously collected and that they are constantly available. However, considering real-world scenarios, this assumption is not completely realistic.

Indeed, in order to be collected, several high-level context data (e.g., semantic location) require interaction with external web services. Continuous network communication may negatively impact the device's resources and latency (i.e., context information is not perfect in real-time).

However, it is important to point out that such high-level contexts do not change so rapidly, while activity recognition is continuously performed every few seconds (e.g., in our experiments, the segmentation window is 4 seconds). Hence, it is possible to design a strategy to obtain new information from web services with a low number of web service calls. For instance, considering semantic location, it

is possible to perform a query only when GPS data exhibit significant changes. As another example, the weather web service could be queried with a low periodicity (e.g., every hour).

Thanks to these strategies, it is also possible to run our method when the user's mobile devices are not connected to the internet for short periods. However, if the mobile devices are offline for a long time period, the system would consider a limited amount of context information, possibly impacting the recognition rate.

In future work, we will investigate in detail such practical aspects, also considering new strategies to adapt the model based on the Quality of Service.

4.6.2 Generalizability of the approach

In this chapter, we focused on the use of our *symbolic features* method for context-aware HAR. However, we are also interested in understanding if our approach could also be applied in different domains. In general, it could be applied to domains where:

- a portion of input data does not directly reveal high-level context information (e.g., inertial sensors in our domain).
- a portion of input data reveals high-level context information (e.g., GPS in our domain).
- it is possible to use common-sense knowledge to define relationships between context and the classification task.

For instance, considering the autonomous driving domain, reasoning on high-level context data may help in improving the decisions made by analyzing the sensors equipped in the smart car. As another domain example, risk assessment and/or security applications may benefit from context reasoning to improve their decisions.

4.7 Summary

In this chapter, we presented our novel knowledge infusion method for context-aware HAR based on symbolic features. Our results have shown how this ap-

proach addresses the research question **Q2** presented in Section 2.5. Indeed, the use of symbolic features mitigates data scarcity in context-aware HAR applications, while being more robust in the presence of noisy context data compared to existing NeSy methods that rely on symbolic reasoning only after the training process of the DL classifier. Moreover, we have shown how the improvements led by probabilistic ontologies do not justify the significant effort required to build them. Like *context refinement*, one of the main limitations of our knowledge infusion approach based on symbolic features is that ontological reasoning is required during classification. This setting may be not suitable for real-world deployments on mobile devices due to the computational complexity of ontologies. In the next chapter, we introduce another knowledge infusion methodology for context-aware HAR based on a semantic loss function that infuses knowledge constraints in the DL classifier only during training, thus avoiding ontological reasoning after deployment.

Chapter 5

Knowledge infusion through a semantic loss function for context-aware HAR

5.1 Introduction

In Chapter 4, we have seen how our method based on the infusion of symbolic features is able to handle data scarcity while being more robust in the presence of noisy context data than *context refinement*, i.e., the state-of-the-art NeSy method for context-aware HAR.

However, both NeSy methods require symbolic reasoning procedures each time an activity prediction is required. In real-world deployments, where the DL classifier can be deployed on resource-constrained machines like mobile/wearable devices, the adoption of symbolic reasoning during classification is not desirable since it is computationally demanding. Indeed, experimental work in the literature shows that running symbolic reasoning on Android mobile devices is up to 150 times slower than on machines with higher resources (e.g., servers) on the considered datasets [25]. In the HAR domain, the work in [1] reports that context-aware ontological reasoning on mobile devices takes, on average, 1.3 seconds for each data sample. This is due to the computational complexity of symbolic reasoners. Considering standard reasoners based on OWL2 ontologies

(that is the most common approach considered in the HAR field [1]), reasoning tasks have polynomial complexity [163]. Hence, even if theoretically considered tractable, these methods do not scale linearly with the size of the knowledge model (e.g., number of activities, context situations, and constraints) and may not be adequate for resource-constrained devices. On the other hand, probabilistic symbolic reasoners like the ones based on log-linear description logics [154] have even higher complexity. While there are approximated methods to reduce the complexity, probabilistic symbolic reasoning is still computationally demanding. Since low-level activities are typically detected with high periodicity (e.g., every few seconds), such approaches may be inefficient in terms of computational resources and energy consumption. Indeed, running the recognition model directly on mobile/wearable devices is a desirable aspect when real-time recognition is a requirement for two main reasons: (1) continuously transmitting sensor signals to a service provider can result in increased latency [164], and (2) onboard sensor processing may be preferred for privacy concerns since such data may reveal sensitive information like personal habits or health conditions [165]. Hence, we believe that removing symbolic reasoning from mobile applications is beneficial.

For this reason, in this chapter we propose a novel Knowledge Infusion method based on a semantic loss function that infuses knowledge constraints in the DL model only during training, thus avoiding symbolic reasoning after deployment. More specifically, we propose a custom loss function that combines a standard classification loss with a novel semantic loss function. The semantic loss component uses symbolic reasoning to drive the DL model in classifying activities considering domain knowledge constraints. After training, the classifier internally encodes such constraints and exploits them at run-time to classify activities without requiring symbolic reasoning.

Our results on the DOMINO [24] and the ExtraSensory [15] datasets show how our *semantic loss* method outperforms in terms of recognition rates a purely data-driven DL approach based on a standard classification loss. Moreover, *semantic loss* often reaches recognition rates close (and sometimes better) to *symbolic features* and *context refinement*, while avoiding the significant cost of performing

symbolic reasoning during inference. Furthermore, our results demonstrate that our *semantic loss* surpasses the other two NeSy approaches in addressing uncertainty, showing significantly greater robustness in the presence of noisy data. Hence, we believe that our semantic loss reaches a good trade-off between efficiency and recognition rate.

The rest of the chapter is organized as follows. Section 5.2 presents our novel knowledge infusion method based on a semantic loss. Section 5.3 describes the experimental evaluation and the results obtained on DOMINO and ExtraSensory. Finally, Section 5.4 compares the strengths and weaknesses of NeSy methods, discusses how to handle cases in which the knowledge model needs to be revised/updated, and presents an initial investigation of the interpretability benefits provided by our semantic loss.

5.2 Knowledge infusion through semantic loss

In this section, we present our novel approach named *knowledge infusion through semantic loss* (or *semantic loss* for short) aimed to overcome the main drawbacks of *symbolic features* and *context refinement*. Our method generates an activity classifier encoding knowledge-based constraints without requiring symbolic reasoning during the inference phase. Hence, a model based on *semantic loss* can be trained offline on a cloud-based server and then deployed on the users' mobile/wearable device to locally perform real-time activity recognition efficiently.

5.2.1 Methodology

In the following, we describe the mechanisms of our *semantic loss* approach, based on the formalism introduced in Section 4.2. As depicted in Figure 5.1, the goal of *semantic loss* is to exploit the knowledge model K to guide the learning process of DNN through a specifically designed loss function. As in the *symbolic features* method, DNN still learns the correlations between context-consistent activities and input data. At the same time, since no additional features are infused into DNN, the use of K and SR during classification is not required,

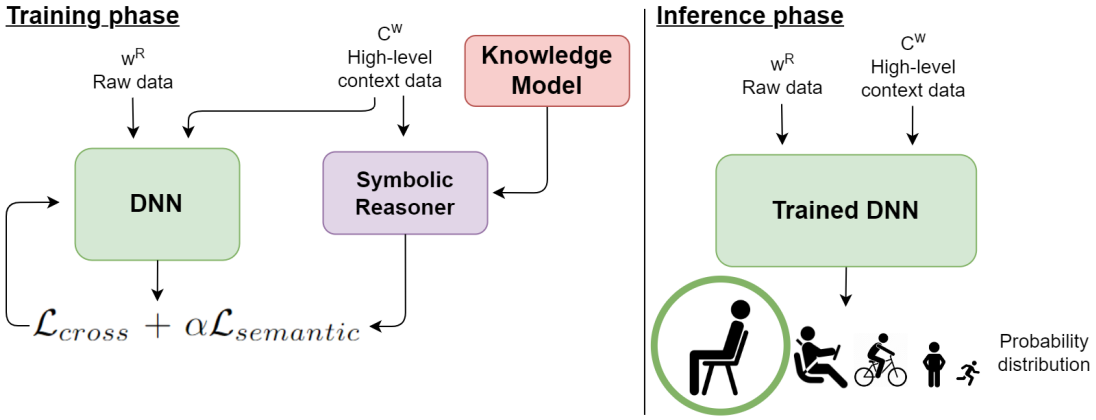


Figure 5.1: Our neuro-symbolic approach based on semantic loss

thus solving one of the main limits of the existing NeSy solutions.

Specifically, the loss function $\mathcal{L} = \mathcal{L}_{cross} + \alpha\mathcal{L}_{semantic}$ that guides the training process of DNN is a combination of the cross-entropy loss function $\mathcal{L}_{cross}$ with a semantic loss function $\mathcal{L}_{semantic}$.

We consider the standard formula for the cross-entropy loss:

$$\mathcal{L}_{cross} = -\sum_{i=1}^{k} y_i \log(p_i) \tag{5.1}$$

where y_i is 1 only when a_i is the ground truth activity, while p_i is the probability of a_i obtained by the DNN.

Consistently with other works in the DL literature [131, 166], α is a trade-off parameter in charge of balancing the different loss terms. In particular, $\mathcal{L}_{semantic}$ determines the impact of the common-sense knowledge about context consistency on the DNN's output.

More formally, let $P = \langle p_1, \ldots, p_k \rangle$ be a probability distribution emitted by DNN on a tuple $\langle w^R, C^w \rangle$, and $l(a_i)$ be the likelihood value obtained by $SR(K, C^w)$ on the activity a_i. We denote with $\hat{p} \in P$ the maximum probability value of P, and with $\hat{a} \in A$ its corresponding activity.

In the following, we describe five alternative semantic loss functions we designed and tested for this thesis.

1. The *AllConsistentActs (All)* semantic loss focuses on the whole probability

distribution P. Intuitively, given P, this semantic loss has the objective of training the network to maximize the sum of the probability values in P that correspond to the context-consistent activities according to $SR()$ (i.e., the ones with likelihood greater than zero). Hence, we would expect that DNN learns to emit non-zero probabilities only for context-consistent activities during classification. Equation 5.2 formally defines the *All* semantic loss:

$$\mathcal{L}_{semanticAll}(P, SR) = 1 - \sum_{i} p_i \cdot l(a_i) \tag{5.2}$$

Since it aggregates probability values with a sum, a potential drawback of this strategy is that different combinations of these values may lead to the same penalty. Hence, the resulting penalties could be poorly informative for DNN to properly learn knowledge constraints. For this reason, the following alternative semantic losses only focus on the most likely activity $\hat{a}$.

2. The *MinusProb-Prob (-PP)* semantic loss aims at associating low probability values with context-inconsistent activities and higher probability values with context-consistent activities. In particular, context-inconsistent predictions are penalized by their probability value. On the other hand, the penalty of context-consistent activities is inversely proportional to the probability $\hat{p}$ of the most likely activity according to the DNN, scaled by the likelihood $l(a_i)$ provided by SR. More formally,

$$\mathcal{L}_{semantic-PP}(P, SR) = \begin{cases} 1 - (\hat{p} \cdot l(\hat{a})) & \text{if } l(\hat{a}) > 0 \\ \hat{p} & \text{otherwise} \end{cases} \tag{5.3}$$

However, a potential drawback of this strategy is that penalty values for consistent activities with relatively low probability values are similar to penalty values for context-inconsistent activities with relatively high probability values.

3. The goal of the *Zero-One (01)* semantic loss is to maximize the differences between penalties of context-consistent and context-inconsistent activities.

Specifically,

$$\mathcal{L}_{semantic01}(SR) = \begin{cases} 0 & \text{if } l(\hat{a}) > 0 \\ 1 & \text{otherwise} \end{cases} \tag{5.4}$$

The following strategies are refined versions of the *01* loss.

4. The *MinusProb-One (-P1)* semantic loss aims at improving the confidence of DNN on context-consistent predictions. Indeed, while the penalty for context-inconsistent activities is fixed, the penalty for context-consistent activities is inversely proportional to the probability $\hat{p}$ of the most likely activity according to the DNN, scaled by the likelihood $l(a_i)$ provided by SR. Hence, context-consistent activities with low probability and/or likelihood values are penalized as well. More formally,

$$\mathcal{L}_{semantic-P1}(P, SR) = \begin{cases} 1 - (\hat{p} \cdot l(\hat{a})) & \text{if } l(\hat{a}) > 0 \\ 1 & \text{otherwise} \end{cases} \tag{5.5}$$

5. Finally, the idea of the *Zero-Prob (0P)* semantic loss is that context-consistent activities should not be penalized, while context-inconsistent activities should be penalized directly proportionally to their associated probability values. Hence, DNN should better learn that the higher the probability values of context-inconsistent activities, the higher the penalty. Therefore, *0P* aims at reducing the probability values on context-inconsistent activities. More formally,

$$\mathcal{L}_{semantic0P}(P, SR) = \begin{cases} 0 & \text{if } l(\hat{a}) > 0 \\ \hat{p} & \text{otherwise} \end{cases} \tag{5.6}$$

In the following, we report a simplified running example of our *semantic loss* approach:

Example 5.1 *A service provider trains, in a supervised way, an activity classifier using a labeled dataset and a symbolic reasoner based on a standard ontology. In particular, each window is fed-forward to the DNN. A loss function combining cross-entropy and AllConsistentActs is used to adjust the weights of the*

DNN. Suppose that, when feed-forwarding a window $\langle w^R, C^w \rangle$*, the output probability distribution of the DNN is the following: Walking:* 50%*, Sitting:* 30%*, Standing:* 15%*, Running:* 5%*. Consider that the ground truth activity is Sitting and that the high-level context* C^w *encodes the information that the current speed is* 0*. By processing* C^w*, the symbolic reasoner outputs the likelihood values for each activity, where Walking and Running have value* 0 *(since they can not be performed with null speed), while the remaining activities have value* 1*. Hence, by applying the formula in Equation 5.2, the value of the semantic loss is* $1-(0.5\cdot 0+0.3\cdot 1+0.15\cdot 1+0.05\cdot 0) = 0.55$*. On the other hand, since the most likely activity does not correspond with the ground truth, the cross-entropy will generate* ≈ 1.73 *as a value. Supposing that* $\alpha = 5$*, the final value of the custom loss is* $1.73+5\cdot 0.55 = 4.48$*, and it will be used to update the weights of the DNN. Hence, the knowledge constraints have a significant impact on determining how to update the weights of the DNN. After training, only the trained classifier is deployed on Alice's smartphone to recognize her activities in real time. Suppose that Alice is sitting, and the smartphone collects a window* $\langle w^R, C^w \rangle$ *of raw sensor data and high-level context data during the execution of this activity. The high-level context* C^w *encodes the information that Alice's current speed is* 0*. By providing the window as input to the activity classifier, it will rely on the knowledge infused during training to assign a high probability to the sitting activity.*

5.3 Experimental evaluation

In this section, we describe the experimental evaluation that we carried out to assess the quality of our method based on a semantic loss. First, we introduce the experimental setup: the datasets we considered, how we pre-processed them, the models used, and the evaluation methodology adopted. Finally, we present the results of our evaluation.

5.3.1 Experimental setup

To evaluate *semantic loss*, we used the same experimental setup considered to evaluate *symbolic features* in Section 4.5.

As datasets, we considered DOMINO [24] and ExtraSensory [15]. To these datasets, we applied the same pre-processing steps already presented in Section 4.5.2. We considered the same DNN architecture used to evaluate *symbolic features*. For the experiments of this chapter, we used the DNN architecture in four different ways:

- As a purely data-driven *baseline*, without further modifications
- Enhanced with our semantic loss (see Section 5.2)
- Enhanced by combining in the *concatenation layer* the *symbolic features* and the features automatically extracted from input data (see description of our *symbolic features* method in Section 4.3)
- As the DNN module of the *context refinement* approach (see Section 4.2.3)

Also in this case, we consider both the standard ontology presented in Section 4.4.2 and the probabilistic ontology described in Section 4.4.3. Finally, we evaluated the different approaches by adopting the same *leave-k-users-out* cross-validation technique presented in Section 4.5.2.

5.3.2 Results

In the following, we show how our *semantic loss* approach outperforms a purely data-driven classifier in terms of recognition rate both in scripted and in-the-wild scenarios. We also compare our method with *symbolic features* and *context refinement*. Our main results consider the standard ontology as the knowledge model since it is a widely used knowledge and context representation framework. The results using an experimental probabilistic knowledge model are presented in Section 5.3.2.

Although *semantic loss* does not include symbolic reasoning during classification, it often reaches recognition rates that are close (and sometimes better) to the ones of the other approaches, especially considering the more realistic scenarios of *ExtraSensory*.

Table 5.1: Comparison between the Semantic Loss types on the different datasets

	Dataset (training set percentage) DOMINO (100%)	ExtraSensory (10%)
Baseline	0.9024	0.5199
MinusProb-Prob (-PP)	0.9139 $\alpha = 5$	0.5402 $\alpha = 4$
Zero-One (01)	0.9042 $\alpha = 1$	0.5270 $\alpha = 7$
Zero-Prob (0P)	0.9162 $\alpha = 3$	0.5288 $\alpha = 9$
AllConsistentActs (ALL)	0.9094 $\alpha = 1$	**0.5872** $\alpha = 30$
MinusProb-One (-P1)	**0.9261** $\alpha = 7$	0.5298 $\alpha = 5$

Semantic loss types comparison

Table 5.1 compares the recognition rates (in terms of overall macro F1 score) of the five semantic loss functions presented in Section 5.2 on *DOMINO* and *ExtraSensory*. To better emphasize the differences in the recognition rates, on *ExtraSensory* we decided to show the results obtained by considering a data scarcity scenario in which only 10% of the training data are available. Indeed, the number of training samples in *DOMINO* is nearly equal to the number contained in only 10% of the training samples in *ExtraSensory*. Moreover, Table 5.1 also includes the best α value for each semantic loss type[1] and the results obtained by the purely data-driven *baseline* that is based on a standard classification loss.

Each semantic loss strategy leads to an improvement in the recognition rates compared to the *baseline*, with *-P1* achieving the best improvements on *DOMINO* ($\approx +2.5\%$) and *All* on *ExtraSensory* ($\approx +6.5\%$). Before running the experiments, we expected similar results for *01*, *-P1*, and *0P* since all these strategies aim at maximizing the distance in penalties between consistent and not-consistent activities. While this insight is confirmed on *ExtraSensory*, on *DOMINO* the *01*

[1] α values have been determined empirically by performing a grid search in the range $[1, 35]$

approach proved to be not very effective in improving the recognition rate. On this dataset, we observed that, besides increasing the difference between the penalties applied to context-consistent and context-inconsistent predictions, it is also crucial to consider the probability values emitted by DNN, especially in the case of a context-consistent prediction, as proved by the *-P1* semantic loss. Finally, the improvement of the *All* strategy on *DOMINO* is limited, probably because learning knowledge constraints considering the whole probability distribution is unnecessarily too hard on simple scripted scenarios. On the other hand, this strategy significantly outperforms the others in the more realistic settings included in *ExtraSensory*.

Comparison with other approaches

Tables 5.2 and 5.3 compare our best *semantic loss* method (i.e., *-P1* on *DOMINO* and *All* on *ExtraSensory*) with: i) the purely data-driven *baseline*, ii) the *symbolic features* strategy, and iii) the *context refinement* strategy. More specifically, we considered different percentages of available training data for each dataset, thus comparing the approaches in different data scarcity scenarios. Note that, during the experimental evaluation, we empirically determined the optimal α values of the *semantic loss* for each training set percentage.

Overall, on each dataset, the NeSy approaches outperform the *baseline*, considering almost all the data scarcity scenarios. This result confirms that traditional symbolic AI approaches have the potential to enhance the predicting capabilities of purely data-driven deep learning models.

Focusing on the scripted scenarios of *DOMINO* (Table 5.2), the improvement of the *semantic loss* is lower than the other approaches, especially considering data scarcity scenarios. For instance, considering 10% of training data, *semantic loss* leads to a recognition rate boost over the *baseline* of $\approx +2\%$ on *DOMINO*. On the other hand, *symbolic features* and *context refinement* lead to improvements of $\approx +13\%$ and $\approx +22\%$, respectively. These performance differences become progressively smaller while increasing training data availability. Indeed, when all the available training data are considered, both *semantic loss* and *context refinement* outperform the *baseline* by $\approx +2\%$, while *symbolic features* leads to

Table 5.2: DOMINO: Results in terms of macro F1 score and 95% confidence interval

Training set percentage	**Baseline**	**Semantic loss -P1**	**Symbolic features**	**Context refinement**
10%	0.5946 (±0.008)	0.6144 (±0.024) $\alpha = 7$	0.7268 (±0.008)	**0.8192** (±0.009)
20%	0.7529 (±0.010)	0.7712 (±0.004) $\alpha = 8$	0.8590 (±0.012)	**0.8811** (±0.007)
30%	0.8268 (±0.006)	0.8469 (±0.002) $\alpha = 9$	**0.9107** (±0.011)	0.9078 (±0.009)
40%	0.8556 (±0.011)	0.8679 (±0.010) $\alpha = 7$	0.9152 (±0.009)	**0.9178** (±0.005)
50%	0.8835 (±0.011)	0.8892 (±0.006) $\alpha = 7$	0.9198 (±0.011)	**0.9281** (±0.012)
60%	0.8917 (±0.010)	0.8889 (±0.007) $\alpha = 7$	0.9237 (±0.009)	**0.9305** (±0.006)
70%	0.8915 (±0.006)	0.9049 (±0.006) $\alpha = 8$	**0.9265** (±0.004)	0.9225 (±0.004)
80%	0.9007 (±0.007)	0.8997 (±0.003) $\alpha = 7$	0.9254 (±0.008)	**0.9274** (±0.005)
90%	0.8965 (±0.002)	0.9021 (±0.008) $\alpha = 6$	**0.9277** (±0.007)	0.9232 (±0.002)
100%	0.9024	0.9261 $\alpha = 7$	**0.9408**	0.9221

Table 5.3: ExtraSensory: Results in terms of macro F1 score and 95% confidence interval

Training set percentage	**Baseline**	**Semantic loss All**	**Symbolic features**	**Context refinement**
1%	0.3127 (±0.023)	0.3366 (±0.027) $\alpha = 29$	0.3418 (±0.010)	**0.6324** (±0.014)
2.5%	0.4279 (±0.008)	0.4895 (±0.010) $\alpha = 30$	0.4720 (±0.016)	**0.6540** (±0.003)
5%	0.4867 (±0.013)	0.5256 (±0.016) $\alpha = 26$	0.5877 (±0.025)	**0.6797** (±0.003)
7.5%	0.5167 (±0.016)	0.5650 (±0.016) $\alpha = 26$	0.6359 (±0.008)	**0.6656** (±0.004)
10%	0.5199 (±0.011)	0.5872 (±0.014) $\alpha = 30$	0.6534 (±0.012)	**0.6622** (±0.007)
25%	0.5842 (±0.016)	0.6331 (±0.013) $\alpha = 29$	0.6404 (±0.010)	**0.6483** (±0.010)
50%	0.6096 (±0.007)	**0.6323** (±0.011) $\alpha = 18$	0.6216 (±0.007)	0.6258 (±0.007)
75%	0.5813 (±0.032)	0.6131 (±0.011) $\alpha = 16$	**0.6268** (±0.007)	0.6067 (±0.023)
100%	0.6053	**0.6244**	0.6205	0.6190

an improvement of $\approx +4\%$.

On the other hand, different insights are observed when focusing on the realistic scenarios of *ExtraSensory* (Table 5.3). Indeed, on this dataset, the differences between the three NeSy approaches are smaller. For instance, considering 10% of training data, the recognition rate improvements of *semantic loss*, *symbolic features*, and *context refinement* are $\approx +7\%$, $\approx +13\%$, and $\approx +14\%$, respectively.

In general, the *semantic loss* achieves improvements that lie between $\approx +2\%$ and $\approx +7\%$, sometimes outperforming the recognition rates of the other NeSy techniques. Indeed, the *semantic loss* outperforms *context refinement* from 50% to 100% of training data, and it also outperforms *symbolic features* on 100% of training data. Overall, *context refinement* is more effective than methods based on knowledge infusion (i.e., *symbolic features* and *semantic loss*) when the availability of labeled data is drastically low. However, when slightly more training data are available (e.g., 25% on *ExtraSensory*), all the NeSy approaches lead to similar improvements.

Our results indicate that our *semantic loss* is effective in capturing relationships between high-level context data and activities with respect to learning them directly from the training set by using purely data-driven models. This is especially true on the *ExtraSensory* dataset, where the improvement of *semantic loss* compared to the *baseline* is larger. Indeed, *DOMINO* covers a significantly lower variability of context situations compared to *ExtraSensory*, and the relationships between context and activities can be captured more easily by the *DNN*. On the other hand, the in-the-wild nature of *ExtraSensory* implies a significantly more complex learning task that can be partially lightened by knowledge reasoning.

Since the computational complexity of symbolic reasoning is not adequate for real-world deployment on resource-constrained devices like smartphones and smartwatches, the choice of the optimal solution should consider a trade-off between recognition rate and efficiency. We believe that our *semantic loss* method is a much more promising approach since it still improves the *baseline* while not requiring symbolic reasoning at all after training.

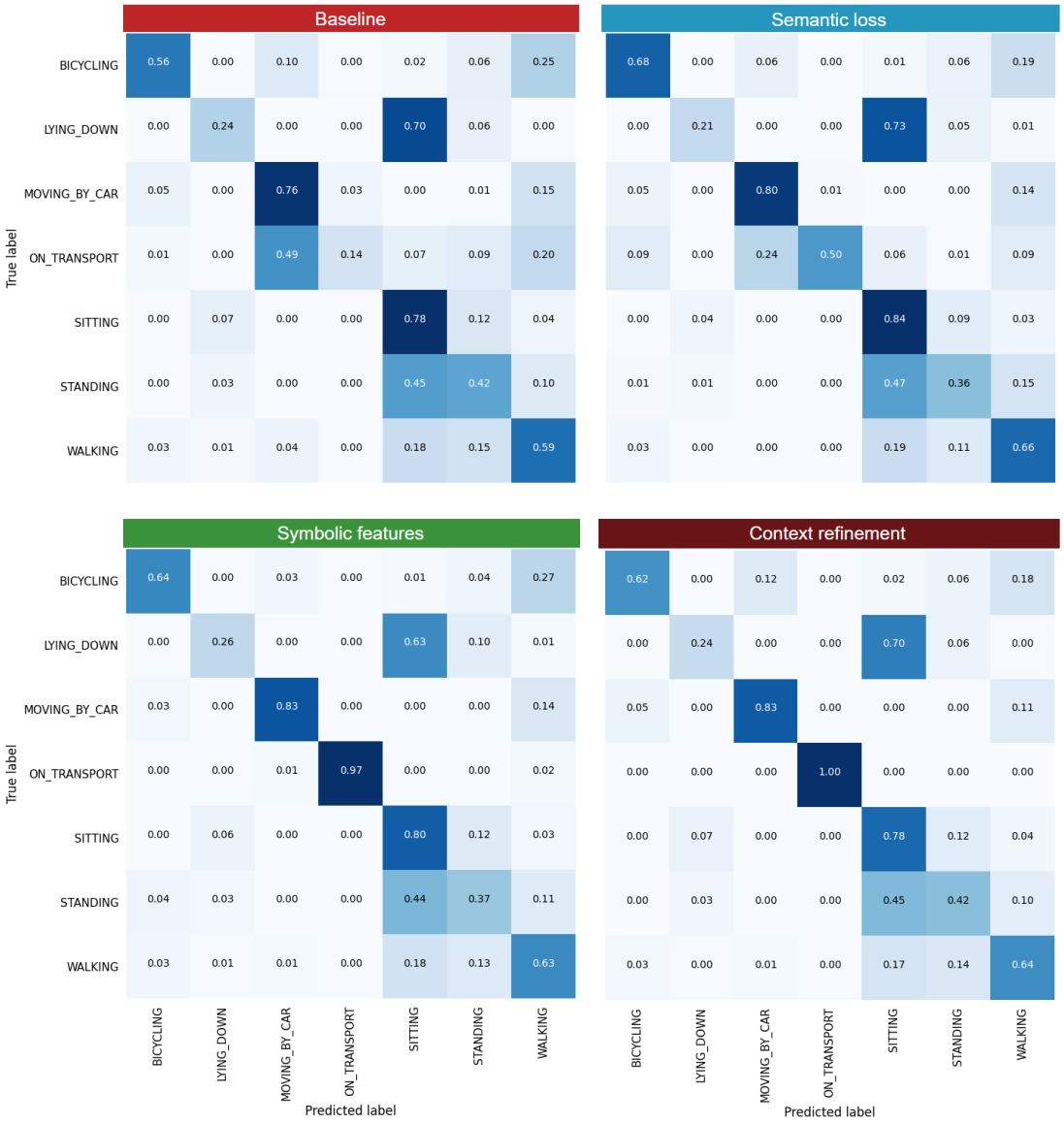


Figure 5.2: Comparison between the confusion matrices of the *baseline* and the three considered Neuro-Symbolic AI approaches trained with 10% of training data on the *ExtraSensory* dataset

Activity-level results

Figure 5.2 compares the confusion matrices obtained by the three considered NeSy approaches and the baseline on *ExtraSensory*, considering the data scarcity

scenario where only 10% of training data are available[2]. From these confusion matrices, it emerges the contribution of domain knowledge in improving the recognition of different activities. For instance, the *baseline* often confuses *on transport* with *moving by car* due to their similar patterns (in terms of inertial measurements and speed), even though context information (e.g., whether the user is following a public transportation route) should help in distinguishing them.

Indeed, even though high-level context data are provided as input to the *baseline*, it is complex to learn from a small training set all the possible correlations between all the possible context conditions and the performed activities. Hence, enhancing the data-driven model with symbolic AI approaches based on domain knowledge has a key role in enhancing the capabilities of the deep learning model and mitigating this problem, thus significantly reducing the confusion between these two activities.

Finally, we observed that each approach performed poorly on the *lying down* activity, which was often confused with *sitting*. We noticed that it is consistent with other papers in the literature that used the ExtraSensory dataset [157]. This is likely due to the fact that both *lying down* and *sitting* are static activities with similar sensor patterns, hence the exact posture is difficult to recognize. Moreover *sitting* is over-represented in the dataset, while *lying down* is underrepresented. For these reasons, the model often outputs *sitting* even if the correct activity is *lying down*.

Robustness to noise

In order to show that our semantic loss is robust to uncertainty, also for this chapter, we performed a set of experiments by introducing noise in the test data. Like we did in Chapter 4, we performed different experiments considering 5%, 10%, and 15% of noisy data in the test set. For each perturbed data sample, we simulated noise in GPS readings, by modifying the semantic location context with another one (plausibly not too distant from the real one) that the knowledge model considers inconsistent with the ground truth activity.

Table 5.4 shows the results of this experiment. We observe that our semantic

[2]We show a representative run among the 5 repetitions of the experiment.

Table 5.4: Average results with 5 different runs in terms of macro f1 score, considering 10% of training data and different percentages of dirty samples in the test set

	Original test set	**5% of dirty test set (delta)**	**10% of dirty test set (delta)**	**15% of dirty test set (delta)**
Baseline	0.5199	0.5089 (- 1.10%)	0.4983 (- 2.16%)	0.4954 (- 2.45%)
Semantic loss	0.5872	**0.5566** (- 3.06%)	**0.5229** (- 6.43%)	**0.5196** (- 6.76%)
Symbolic features	0.6534	0.5498 (- 10.36%)	0.5200 (- 13.34%)	0.5043 (- 14.91%)
Context refinement	**0.6622**	0.5430 (- 11.92%)	0.5057 (- 15.65%)	0.4801 (- 18.21%)

loss is significantly more robust to noise compared to the other NeSy methods while outperforming the *baseline* in each considered setting. For instance, considering 5% of noisy data samples, *semantic loss* presents a decrease in the macro F1 score of only -3.06% compared to the -10.36% of *symbolic features* and the -11.92% of *context refinement*.

Results with a probabilistic knowledge ontology

Table 5.5 summarizes the results that we obtained on both datasets by using a probabilistic knowledge model slightly adapted from the one proposed in [64]. For

Table 5.5: Average results with 5 different runs in terms of macro F1 score, considering a data scarcity scenario simulated by using 10% of training data and the probabilistic version of each method

	DOMINO		**ExtraSensory**	
	Standard	Probabilistic	Standard	Probabilistic
Baseline	0.5946	0.5946	0.5199	0.5199
Semantic loss	0.6144	**0.6372**	0.5872	**0.6013**
Symbolic features	0.7268	**0.7365**	**0.6534**	0.6408
Context refinement	0.8192	**0.8399**	0.6622	**0.6793**

the sake of simplicity, we show the results considering the data scarcity scenario

where only 10% of labeled data are available. Our results confirm that, also for the *semantic loss* approach, introducing fuzziness only slightly improves the recognition rates obtained with a standard ontology. We believe that this does not justify the effort of designing and managing probabilistic ontologies.

5.4 Discussion

5.4.1 Strengths and weaknesses of Neuro-Symbolic approaches

In the following, we discuss the strengths and weaknesses of the three Neuro-Symbolic AI (NeSy) approaches compared in this chapter: *context refinement*, *symbolic features*, and *semantic loss*. This information is also summarized in Table 5.6.

Table 5.6: Comparison of pros and cons of NeSy methods

	context refinement	symbolic features	semantic loss
improving recognition rate	x	x	x
mitigating data scarcity	x	x	x
retraining not required when knowledge is revised	x		
handling data uncertainty			x
symbolic reasoning not required after deployment			x

Compared to other methods, *context refinement* often reaches the highest recognition rates, especially when the amount of available training data is limited. However, this method may be less effective when based on an imperfect knowledge model. Indeed, *context refinement* always discards activities only relying on the user's surrounding context considering rigid constraints. For instance, a user could ride a bicycle even in unusual context scenarios (e.g., on a pedestrian-only road). Hence, when the knowledge model does not cover all the possible contexts in which an activity can be performed, combining the information from inertial data with knowledge would be more convenient in refining the probability

distribution. Moreover, our results show that context refinement performs poorly in the presence of uncertainty in context data.

While the *symbolic features* method is less accurate than *context refinement*, it is slightly better in capturing the intrinsic uncertainty in sensor data by learning correlations between features and contexts, as opposed to the latter's direct application of rigid rules.

However, both approaches require the use of the symbolic reasoning module at each activity prediction, making them less suitable for deployment on mobile devices. Moreover, both approaches are significantly less effective than semantic loss in the presence of uncertainty in context data.

On the other hand, our *semantic loss* can be trained offline on a server with high computational capabilities and then deployed and used on a mobile device without the need for computationally expensive symbolic reasoning tasks. Indeed, *semantic loss* is still able to significantly improve the recognition rate. Additionally, it is the most robust NeSy approach when context data is noisy.

5.4.2 Revising/updating the knowledge model

In this work, we assumed that the knowledge model is static and never updated. However, this is not necessarily true in real-world settings. Indeed, we expect that the model can be *extended* by including new knowledge and/or *revised*.

If the knowledge is *extended* by including new activities or new context sources, all the NeSy models have to be modified to accommodate for new inputs and/or new output classes. New representative training data are also required. On the other hand, the knowledge can be *revised* to refine existing constraints between contexts and activities. For instance, domain experts may realize that the existing constraints are not adequate and should be improved. In this scenario, an advantage of context-refinement is that it does not require retraining the DNN, since symbolic reasoning is applied only during classification. However, re-training is required for the approaches based on knowledge infusion.

In our scenario, the model is pre-trained offline by a service provider with storage and computational capabilities and then deployed on mobile devices for inference. Hence, we believe that in this scenario, the service provider could

easily re-train the model from scratch by taking into account the new knowledge model and possibly new representative data points.

When this is not possible or convenient, we believe that continual learning approaches (e.g., based on the teacher-student paradigm) could be adopted to incrementally train the underlying deep learning model to retain previous knowledge and learn new constraints, without the need for re-training from scratch. We believe that existing continual learning approaches could be effective when the knowledge model is *extended*, while it is more challenging when it is *revised* since incremental learning should allow the model to select which constraints to retain and which to update.

A more in-depth investigation on how to incrementally train neuro-symbolic approaches upon changes in the knowledge model is the subject of future work.

5.4.3 Interpretability

In the literature, Neuro-Symbolic AI methods are well-known for improving the interpretability of deep learning models [52]. Indeed, considering Knowledge Infusion, the decisions of a NeSy model are driven by the infused knowledge. Hence, the knowledge model itself can be used to interpret the output of the classifier. Moreover, eXplainable AI methods (XAI) such as the model induction (e.g., LIME [123]) or the saliency-based ones (e.g., GradCAM [128]) can be used to further inspect how the deep learning model reaches each decision.

In this paper, the knowledge infused into the model is about the relationships between high-level context data and activities. To better inspect the interpretability aspects of our model, we applied an XAI model induction approach named RISE [167] to visualize the importance of high-level context features on the supervised baseline (i.e., without knowledge infusion) and on our semantic loss model[3]. As an example, Figure 5.3 shows the average importance of high-level context features on the *ExtraSensory* dataset for the activity *on transport* on the *baseline* model. Figure 5.4 shows the same result for the semantic loss model.

[3]For this evaluation, we randomly split the dataset into 70% for train, 10% for validation, and 20% for test; the models were trained on the train set and feature importance was computed on the predictions made on the test set.

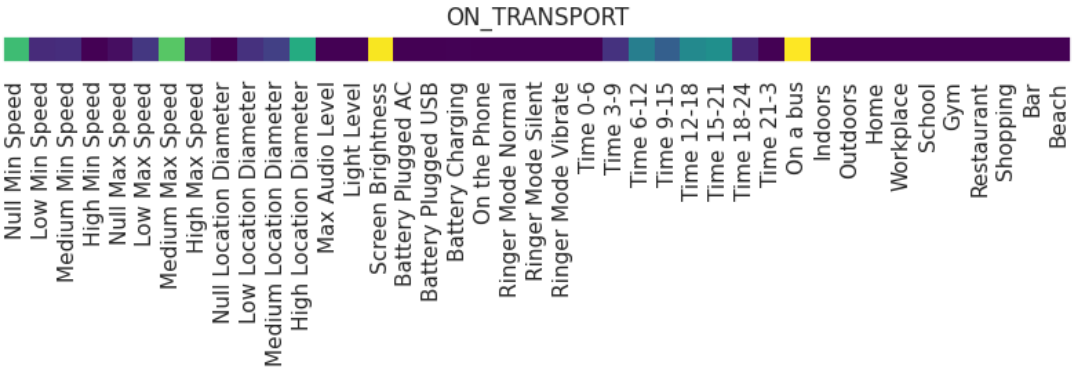


Figure 5.3: Average feature importance for the *on transport* activity obtained using XAI methods on the **baseline** model. The brighter the color, the more important the corresponding feature was for classification.

Figure 5.4: Average feature importance for the *on transport* activity obtained using XAI methods on the **semantic loss** model. The brighter the color, the more important the corresponding feature was for classification.

We observe that the *baseline* model considers important many features that are not directly related to the activity, like *screen brightness*. On the other hand, the *semantic loss* model, consistently with the infused knowledge, considers particularly important only the context *on a bus*. Taking into account our results in Figure 5.2, it is clear that this improvement led the classifier to achieve better results since it focuses on context features that are actually relevant considering the knowledge model.

However, in this work, the classifier's decision is not based only on context data, but also on inertial sensor data that are inherently challenging to explain.

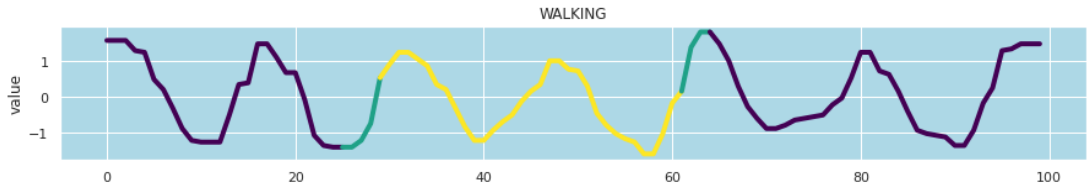


Figure 5.5: Example explanation for a sample of the *walking* activity based on the x-axis measurements from the smartwatch's accelerometer. The brightness of the color indicates the level of importance of each measurement for classification.

While it is possible to highlight the portion of the signal that was important for the classifier (e.g., see Figure 5.5), this is difficult for humans to interpret and our knowledge model does not affect the interpretability of such signals. Therefore, our knowledge infusion approach leads to a deep learning model that is *partially* interpretable.

5.5 Summary

In this chapter, we presented a novel Neuro-Symbolic AI approach for context-aware HAR based on a combination of a standard loss function for classification with a semantic loss. This approach addresses the research question **Q3** presented in Section 2.5. Indeed, the use of semantic loss avoids symbolic reasoning during classification, thus making the model deployment feasible even on devices with limited computational resources. Overall, our results have shown how our method improves the recognition rates of a purely data-driven model. The advantage of our approach is particularly evident in the realistic in-the-wild settings included in the ExtraSensory dataset [15]. Moreover, compared to *context refinement* and *symbolic features*, our *semantic loss* is particularly promising in coping with uncertainty in context data. At the end of this chapter, we have also made an initial qualitative analysis of possible interpretability benefits provided by Knowledge Infusion methods. Unfortunately, in the current sensor-based HAR literature, no quantitative metric has been introduced to measure the interpretability level of DL models. In the next chapter, we propose a novel methodology to apply to sensor data existing eXplainable AI (XAI) methods originally designed for computer vision tasks. In this context, we also introduce an innovative metric

that we designed to measure the coherence of the explanations generated through XAI methods with human knowledge about the HAR domain. This metric can be considered in the future to quantify the interpretability benefits provided by NeSy methods for sensor-based HAR.

Chapter 6

Explainable deep learning classifiers for sensor-based HAR

6.1 Introduction

As we have already discussed in the introduction of this thesis, in the last few years, sensor-based HAR has been mainly tackled with data-driven activity classifiers based on Deep Learning (DL). However, one of the major problems of DL models is their opacity: it is challenging to understand the rationale behind their predictions [116]. Explainable Artificial Intelligence (XAI) approaches recently emerged to address this problem [117]. XAI aims to provide a human-understandable explanation associated with each model's prediction. Applications based on HAR heavily rely on the output of activity recognition frameworks. Hence, inferring why a classifier predicted a specific user's activity is essential to provide solutions that are understandable, trusted, and transparent [118]. For instance, consider a healthcare system that analyzes the daily routines of elderly subjects. The detection of Activities of Daily Living (ADLs) is one of the fundamental steps to detect higher-level behaviors to support clinicians' diagnoses (e.g., cognitive decline) and interventions [119]. XAI would allow clinicians to increase their trust in decision support systems that rely on ADL recognition. Explanations are also useful to data scientists who need to refine the recognition system by introducing, removing, or re-positioning sensors, modifying algorithms

and system parameters, or revising/extending the training set. An explainable system would also make it possible to include residents and caregivers in the loop, by showing them which ADLs are released to clinicians and how the system inferred their execution.

The sensor-based HAR literature has only investigated XAI solutions based on interpretable standard machine learning classifiers [141]. While those models can inherently explain their predictions, they usually provide lower recognition rates than DL, and they require manual feature extraction and selection. Moreover, existing works do not tackle the problem of making explanations understandable by non-expert users (e.g., clinicians and caregivers). Hence, it is still an open problem to understand if and how XAI can be combined with DL-based activity recognition.

In Chapter 5, we made a first step toward analyzing possible interpretability benefits provided by Neuro-Symbolic AI methods for sensor-based HAR with DL models. However, we have just qualitatively observed that Knowledge Infusion may lead to models that are *partially* interpretable. Hence, our initial analysis presents three main limitations. Firstly, it focused on interpreting the model's predictions according only to high-level context data (i.e., input data with a clear semantic). However, DL-based activity classifiers typically make their decisions also relying on raw sensor measurements, which existing XAI methods struggle to take into account since they have been mainly designed for computer vision tasks. Secondly, our analysis was based on explanations generated as heat maps. These explanations can be adequate for data scientists and machine learning experts, but not for non-expert end users. Finally, the use of a quantitative metric could give additional insights into the interpretability benefits of Neuro-Symbolic AI. Unfortunately, in the current sensor-based HAR literature, no quantitative metric has been introduced to measure the interpretability level of DL models.

In this chapter, we try to address these three limitations by considering the recognition of ADLs as the application domain. More specifically, we propose DeXAR: a novel methodology for explainable sensor-based ADL recognition exploiting DL models. DeXAR extracts high-level semantic information from raw sensor data and ADLs previously performed by the resident of a smart home in real-time.

From this information, DeXAR generates semantic images that are processed by a DL classifier providing a prediction about the activity currently performed by the resident. DeXAR obtains an explanation for each prediction generating a heat map that associates a relevance value to each pixel of the input semantic image. Intuitively, the heat map reveals the rationale behind the classification of a semantic image based on what the DL model actually learned during training. We considered three candidate XAI approaches: Grad-CAM as a saliency-based *deep explanation* approach [128], LIME as a *model induction* approach [123], and Model Prototypes as a novel white-box *deep explanation* approach that we adapted from the one proposed in [130]. While a heat map may be very informative for data scientists, it is poorly understandable by non-expert users. Hence, DeXAR also includes a module to transform heat maps into sentences in natural language.

We performed an extensive evaluation of DeXAR on two public datasets of ADLs. We first show that the DL classifiers used by DeXAR reach satisfactory recognition rates. We then evaluate the explanations generated with the three XAI methods through two separate studies. The former adopts a quantitative measure we designed (i.e., the Explanation Score) to evaluate the coherence of the explanations with common-sense knowledge. The latter is composed of two user-centered studies that involved 84, and 63 participants, respectively. Our results on both datasets indicate that our white-box approach based on prototypes provides the best explanations. Moreover, the results indicate that evaluations based on the Explanation Score are aligned and consistent with user-based scores obtained through surveys. This suggests that our metric can be used to quantify how DL models for sensor-based HAR are interpretable for humans.

To the best of our knowledge, this is the first framework that includes XAI approaches based on DL for sensor-based HAR, generating natural language explanations. Moreover, we believe that the Explanation Score introduced in this chapter can be used in the future to evaluate the interpretability benefits of Neuro-Symbolic AI methods for HAR.

The rest of the chapter is organized as follows. Sections 6.2 and 6.3 present DeXAR and describe the experimental setup we considered to evaluate it, re-

spectively. Finally, Section 6.4 discusses the main limitations of DeXAR.

6.2 Methodology

In this section, we describe DeXAR: our approach to take advantage of deep learning and XAI methods to enable explainable ADL recognition. Figure 6.1 shows the overall data flow of DeXAR. For the sake of this chapter, we consider a sensorized smart home with a single resident. Several environmental sensors are deployed in the environment to capture the resident's interaction with the surrounding infrastructure and her location in the home. The resident also wears a wearable device (e.g., a smartwatch) in charge of collecting inertial sensor data to capture her physical movements. In order to enable explainability, we derive a high-level representation of raw sensor data that we call *semantic states*. We perform temporal segmentation on semantic states and we generate a semantic image from each segment. A semantic image encodes the semantic states observed within the time span of the corresponding segment, as well as the latest K ADLs that the resident performed before the current one. The image is processed by an ADL classifier based on deep learning. By applying XAI methods, we associate the output of the classifier with a heat map that indicates the features of the image that were important for classification. Finally, we apply a method that maps the heat map to a semantic explanation in natural language that is prompted to the end-user. In parallel, as we will see in Section 6.3, symbolic reasoning can be used to quantitatively evaluate (through our Explanation Score) the consistency of each explanation with common-sense knowledge about HAR. In the following, we explain each step of DeXAR in detail.

6.2.1 Deriving semantic states from sensor data

In general, XAI makes sense if it is possible to understand the semantics of each explanation. However, sensor-based ADL recognition usually relies on raw data that are difficult to explain. This is especially true considering, for example, inertial sensors that generate continuous values about the physical movements of the user on three axes. Hence, we pre-process the stream of raw sensor data to

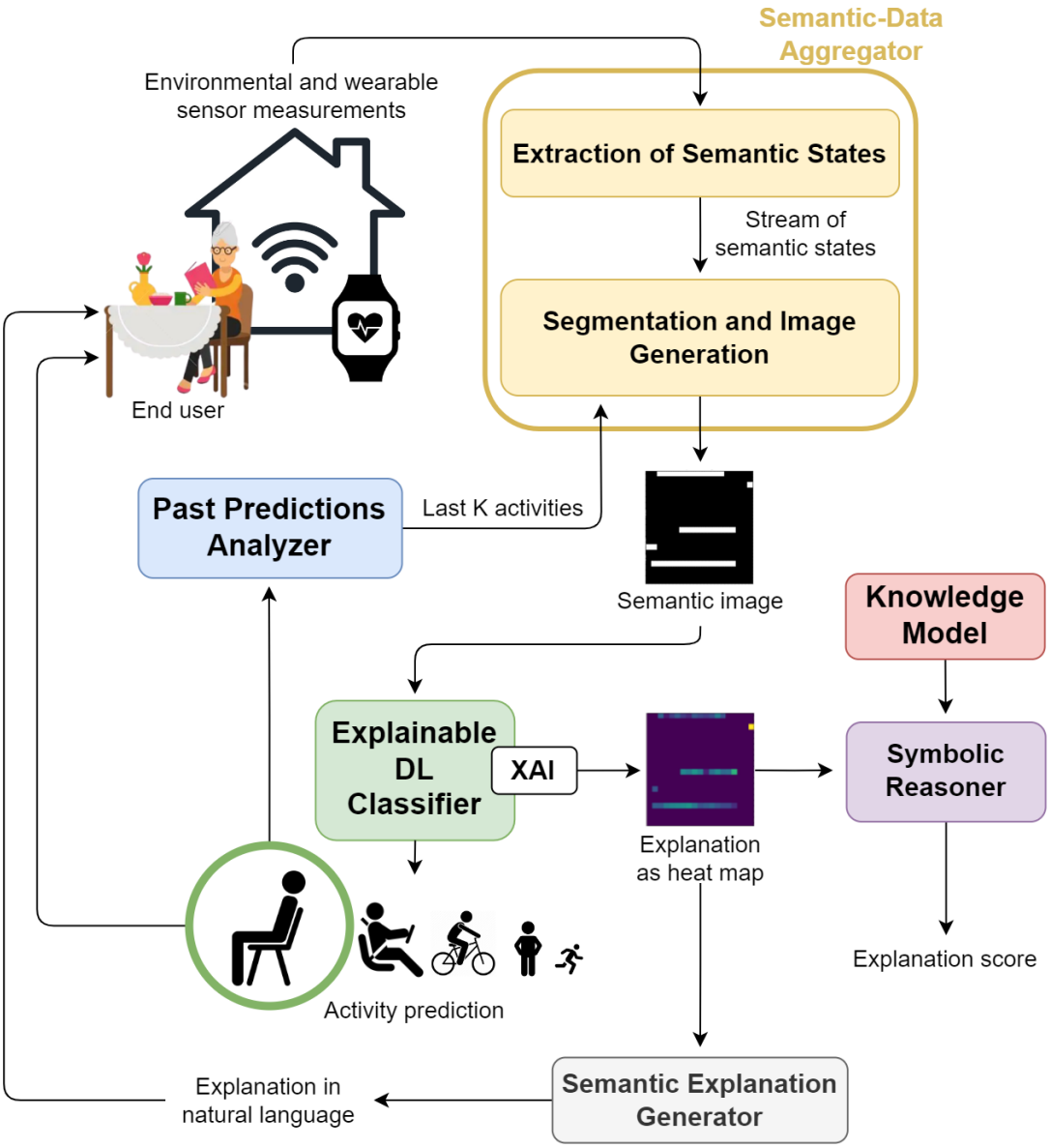


Figure 6.1: The overall data flow of DeXAR

derive a stream of *semantic states*: high-level information with a clear semantic that describes what happened within a time interval. For each sensor s_i, we denote with $\mathbf{S^i}$ the set of types of semantic states that it is possible to derive from raw measurements of s_i. We denote with $S[ts, te]$ a semantic state of type S occurred within an interval $[ts, te]$ with ts and te timestamps and $ts < te$. The ending timestamp of a semantic state may be undefined (in this case we represent

the interval $[ts, -]$) indicating that the semantic state is currently *active*. As we describe in the following, the method used to derive semantic states from sensor data depends on the specific type of sensor.

Environmental sensors

Environmental sensor data capture the resident's interaction with the surrounding home infrastructure. These sensors usually have a binary output: they generate the value ON or the value OFF at specific time instants. For instance, magnetic sensors can detect when doors/drawers are opened and closed. Pressure mat sensors on the chairs can detect when someone is sitting. Plug sensors can detect the usage of home appliances. Raw data from environmental sensors can be easily mapped into semantic states. An environmental binary sensor generates semantic states when it is activated (i.e., when it outputs the ON value).

Example 6.1 *Suppose that the smart home is equipped with a pressure mat sensor identified as P_2 on the kitchen chair. When the resident sits on that chair, the sensor will output the value ON at time t. When our method processes this value, based on the knowledge about the events that P_2 is monitoring, it generates the semantic state Using_kitchen_chair$[t, -]$. Note that the end time of this semantic state is undefined (i.e., the state is currently active). Suppose that subsequently, at a time instant t', the same sensor P_2 outputs the value OFF. In this case, our method, based on the knowledge that this measurement implies that the person who was sitting on the kitchen chair is now standing, updates the state as Using_kitchen_chair$[t, t']$.*

Some environmental sensors (e.g., PIR, BLE beacons) may also reveal the semantic location of the resident in the home during a time interval, hence generating a semantic state like $In\_The\_LivingRoom[t_i, t_j]$.

Note that we assume that, for each environmental sensor deployed in the home environment, it is possible to know the corresponding objects, actions, and location. We believe that this assumption is realistic since it is possible to obtain this information during the deployment of the sensors in the smart home.

Inertial sensors

Mapping raw inertial sensor data to semantic states requires a more sophisticated approach. Consider, for instance, an accelerometer that continuously generates values on three axes at a high frequency. Clearly, it is not possible to directly associate semantics with those values.

Nonetheless, it is possible to use machine learning methods to infer higher-level information from those sensor data. For instance, inertial sensor measurements generated by the sensors equipped on a wristband can be processed by machine learning classifiers to reliably derive simple gestures (e.g., the resident is raising her arm, the arm is still, the resident is performing some manipulation, etc) [168]. Deriving simple physical activities, postures, and gestures from inertial sensors using machine learning is a well-established methodology in the literature [3]. Existing works suggest that the sequences of low-level physical activities, postures, and gestures can reveal higher-level activities [169]. DeXAR relies on such types of machine learning algorithms to reliably derive simple low-level activities from the stream of inertial sensors, mapping the output to semantic states. Note that the specific method being used strictly depends on the available devices (e.g., smartwatches, smartphones) and the low-level activities that are required for the specific application (e.g., postures, hand gestures, physical activities)[1]. We generate a new semantic state when a user switches from a low-level activity a_1 to a low-level activity a_2. For the sake of this chapter, we only consider simple actions that can be reliably classified by existing techniques.

Example 6.2 *Suppose that at time t_j, the user switched its low-level posture from standing to sitting and this is captured by a machine learning classifier from the inertial sensors data of the user's smartphone. Given this situation, our method updates the semantic state related to standing to $Standing[t_i, t_{j-1}]$ and generates a new semantic state related to sitting as $Sitting[t_j, -]$.*

[1] In our experiments, we exploit smartwatches' inertial data to derive simple hand gestures. Details about the specific method are reported in the experimental section.

Global stream of semantic states

Each stream of measurements from a sensor s_i generates a stream of states whose type is in $\mathbf{S}^i$: $State\_Stream(s_i) = \langle S_1^i[ts_1, te_1], S_2^i[ts_2, te_2], \ldots, \rangle$, where $S_1^i, S_2^i, \ldots, \in \mathbf{S}^i$ and $te_1 < ts_2$. Note that the same state type can appear multiple times in the stream. For instance, a magnetic sensor on the medicine drawer generates a state stream like $\langle$*Medicine_drawer_open*$[t1, t2]$, *Medicine_drawer_open*$[t5, t6], \ldots\rangle$. On the other hand, a posture detection classifier on wearable sensors data may generate a state stream like $\langle$*Standing*$[t1, t2]$, *Sitting*$[t3, t4]$, *Standing*$[t5, t6], \ldots\rangle$. By joining the state streams from all the sensors we obtain a *global state stream*: $Global\_State\_Stream = \langle S_1[ts_1, te_1]$, $S_2[ts_2, te_2], \ldots, \rangle$, where $S_1, S_2, \ldots \in \bigcup \mathbf{S^i}$. Please, note that the intervals in this stream, corresponding to states derived from different sensors, can have non-empty intersections. An example of a global state stream is: $\langle$*Standing*$[1, 6]$, *Moving_arm*$[2, 5]$, *Fridge_door_open*$[2, 3], \ldots\rangle$

6.2.2 Segmentation

We use a fixed-length sliding window approach to segment the global stream of semantic states. Each segment only captures what actually happened within its time span. Indeed, each segment only includes the sub-intervals of the semantic states that overlap with the time interval of the segment. More formally, each segment spans n seconds with a factor ov of overlap. Given a segment whose time-span is $[t_i, t_j]$ and a semantic state $S[ts, te]$, let $[t_a, t_b] = [ts, te] \bigcap [t_i, t_j]$ be the intersection of the time-intervals. $S[t_a, t_b]$ is associated with the segment if and only if $[t_a, t_b] \neq \emptyset$ (i.e., the intersection is non-empty).

Example 6.3 *Consider a segment that spans over the interval* $[10, 16]$. *Suppose that the current stream of semantic states is the following:* $\langle$*Standing*$[8, 13]$, *Fridge_open*$[11, 14]$, *PantryDrawer_Open*$[15, -]\rangle$. *In this case, our segment will consider semantic states as follows:* $\langle$*Standing*$[10, 13]$, *Fridge_open*$[11, 14]$, *PantryDrawer_Open*$[15, 16]\rangle$.

6.2.3 Information about past activities

The current activity of the resident may be semantically related to her preceding activities. Consider, for example, the activity *eating*. This ADL often occurs after *cooking* and *setting up the table*. For this reason, considering the information about past activities may contribute both to the recognition of the current activity as well as to generate a more sophisticated explanation.

In the following, we propose a heuristic-based approach to derive reliable information about past activities. We process in real-time the output of the classifier to keep track of *stable activities predictions*. When the system observes a sequence of consecutive segments that are classified with the same ADL A, we generate a *stable prediction* of the activity A if and only if at least t times the prediction confidence on A is higher than a threshold c. We assume that the most recent stable prediction is the current activity of the resident.

In order to classify a segment, we consider the most recent past K stable predictions that the resident performed before the current activity. We combine the segment of semantic states and these K stable predictions to generate the input for the classifier.

In order to reduce overfitting problems, we train our classifier with and without information about past activities. This makes it possible to generate an activity model capable of generalizing independently from specific sequences of activities.

6.2.4 Image generation

In the following, we describe how we obtain an image representation starting from a segment of semantic states and past ADLs. Our image generation approach is inspired by the work proposed in [170], which we adapted to encode our semantic states and the information of past ADLs. Figure 6.2 depicts an example of an image[2] generated by DeXAR. The image has a black background, and white pixels encode information about semantic states and past ADLs. In particular, our image is a binary matrix with shape $(S, n + K + 1)$, where S is the overall

[2]As we will see later, this is actually a binary matrix that can be seen as an image for visualization purposes

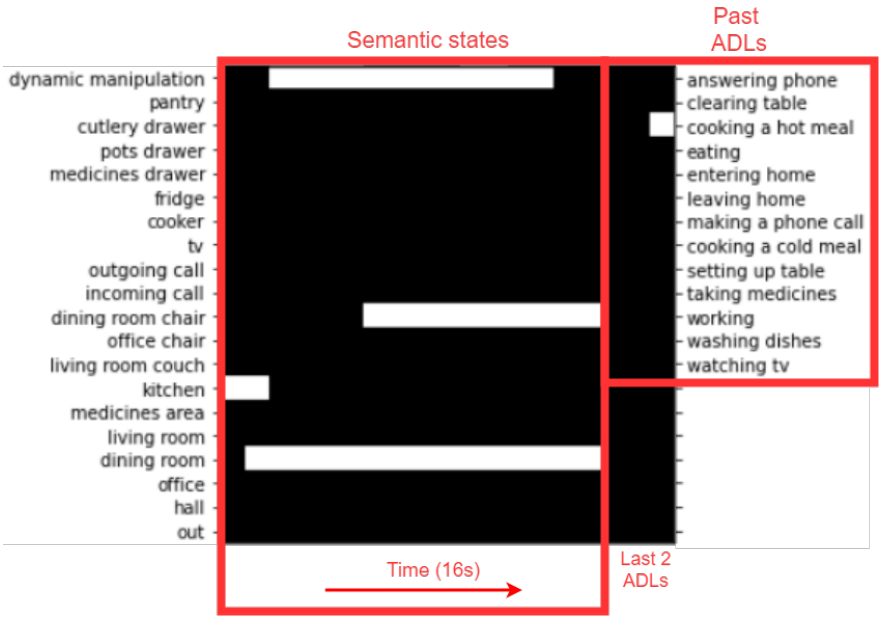


Figure 6.2: An example of an image generated from the MARBLE dataset [2] (more details will be presented in Section 6.3) related to the *eating* activity. The sub-matrix on the left encodes the temporal relationships of semantic states. For instance, in this temporal window of 16 seconds, the user was mostly in the dining room sitting on the dining room chair. Also, he performed some *dynamic* hand gestures, probably to eat. The sub-matrix on the right shows information about past activities. In this case, there is only one previous activity: *cooking a hot meal*

number of semantic state types, n is the segment's length (in seconds), and K is the number of past ADLs. The matrix is actually a combination of two sub-matrices: the former, with shape (S, n), encodes the temporal dependencies between semantic states. The latter, with shape (C, K), encodes information about past ADLs. Here, C is the overall number of activity classes. We assume that $C \leq S$. Note that there is one empty column between the first n columns and the last K columns, to clearly separate semantic states from past activities information.

Considering the semantic states matrix, each row represents a semantic state type and each column represents a specific second within the segment (e.g., column 3 is the third second inside the segment). The value of this matrix at row i and column j is 1 if the semantic state of type S^i (e.g., *Fridge_open*) was active

at second j, 0 otherwise.

Considering the past ADLs matrix, each row represents an activity class. The value of this matrix at row i and column j is 1 if the $(K-j)$-th past ADL was an instance of the activity i. Hence, the temporal order is from left to right (i.e., the right-most column is related to the most recent past activity).

Example 6.4 *Suppose $K = 2$ is the number of past ADLs considered to generate the semantic image and a segment of* 16 *seconds. This segment includes the following semantic states:*

$$\langle In\_The\_Kitchen[0,2], In\_The\_Dining\_Room[2,16],$$
$$Dynamic\_Manipulation[3,14], Using\_Dining\_Room\_Chair[7,16]\rangle$$

The method proposed in Section 6.2.3 derives cooking a hot meal as the only stable past activity. Hence, our image generation algorithm would output the semantic image depicted in Figure 6.2.

To sum up, the semantic states matrix encodes temporal properties of the semantic states including the duration, while the past ADLs matrix simply reports the temporal order of ADLs detected by the classifier in the recent past without any information about their duration.

6.2.5 Deep XAI approaches

As we described above, each pixel of an image generated by DeXAR has a well-defined semantic. Hence, we apply XAI methods to derive which pixels are important for a DL model during classification. We investigated three different categories of XAI approaches: Grad-CAM [128] as a saliency-based deep explanation approach, LIME [123] as a model induction approach, and Model Prototypes [130] as a white-box deep explanation approach. These methods generate a heat map that indicates the influence of each pixel on the classification of a particular semantic image. The higher the intensity, the more important the pixel. Since the original version of the Model Prototypes approach does not actually generate a heat map, we extended it as we will describe in Section 6.2.5.

Grad-CAM

Gradient-weighted Class Activation Mapping (Grad-CAM) [128] is an XAI approach that works on any DL model based on convolutional layers, without requiring architectural changes or re-training. The rationale behind this approach is that the last convolutional layers encode the semantic class-specific information in the image. Grad-CAM analyzes the gradients of the network to derive the importance of each neuron during classification. Based on the features extracted by the convolutional layers, Grad-CAM computes the global average pooling of the gradients that are associated with the predicted class. The output is a heat map where the convolutional features that have a positive influence on the predicted class result in pixels with high intensity.

More formally, given an input img, let $F = \{F^1, \ldots, F^m\}$ be the set of feature maps extracted from the last convolutional layer of the network when classifying img. Moreover, let A be the predicted activity, and y^A the score that the network associates with the activity A just before the softmax layer. For each feature map $F^k \in F$, Grad-CAM derives the *neuron importance weights* α_k^A as follows:

$$\alpha_k^A = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^A}{\partial F_{ij}^k} \tag{6.1}$$

Here, $\frac{\partial y^A}{\partial F_{ij}^k}$ is the gradient of the score y^A with respect to the feature map F^k based on back-propagation, Z is a normalization factor, and i and j represent the row and column indices. Finally, the heat map hm corresponding to img is generated as the output of the ReLU function applied to a weighted combination of the feature maps:

$$hm = ReLU(\sum_k \alpha_k^A F^k) \tag{6.2}$$

Note that the $ReLU$ function excludes from the heat map the pixels with negative intensity, since we are only interested in features that have a positive influence on the class of interest.

Figure 6.3 shows an explanation generated by Grad-CAM on the example image depicted in Figure 6.2 that was classified as *eating*.

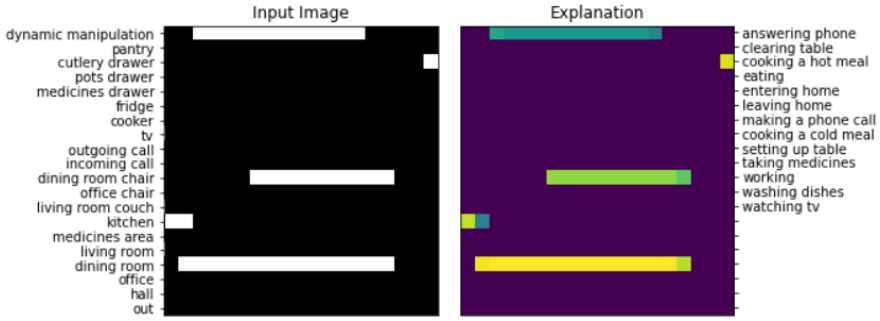


Figure 6.3: Grad-CAM: An example of an explanation for an input related to the *eating* activity. The most important aspects (in yellow) are the location in the dining room and the fact that the user previously performed *cooking a hot meal*. Grad-CAM also finds out that is relatively important (light green) that the subject was previously in the kitchen and that he was sitting on the chair of the dining room. Even though the resident performed hand manipulations, these are associated with low importance

LIME

The Local Interpretable Model-Agnostic Explanations (LIME) [123] approach is based on model induction. LIME can be used to explain the predictions of any classifier, that is considered a black-box. In order to produce an explanation, LIME generates l perturbations of the input to train a sparse linear model using the output of the classifier on these perturbations. The linear model associates a coefficient to each feature of the input. The features (in our case, the pixels of the image) associated with large coefficients in the linear model are considered to be important in explaining the prediction. We apply LIME to obtain a heat map where large coefficients in the linear model result in pixels with high intensity.

More formally, given a black-box model h and an input image img to be explained, LIME generates an interpretable linear model g by solving:

$$argmin_g = \mathcal{L}(h, g, \pi_{img}) + \Omega(g) \tag{6.3}$$

where π_{img} is a proximity function that relies on an exponential kernel based on the L2 distance, $\mathcal{L}$ is a locality-aware loss (i.e., how unfaithful g approximates h in the locality π_{img}), and $\Omega(g)$ is the complexity of the linear model. Note that π_{img} is used to measure the distance between img and another image img'.

LIME approximates $\mathcal{L}$ by generating l perturbed image samples of img weighted by π_{img}. Each perturbed instance img' is obtained by drawing nonzero elements of img uniformly at random. For each perturbed image img', the corresponding label $h(img')$ is computed. Using the perturbed images and the corresponding labels, Eq. 6.3 is minimized to obtain the coefficients w_g of a sparse linear model. Hence, the heat map is computed by weighting the input image img with w_g:

$$hm = w_g \cdot img \tag{6.4}$$

Figure 6.4 shows an explanation generated by LIME on the example image depicted in Figure 6.2 that was classified as *eating*.

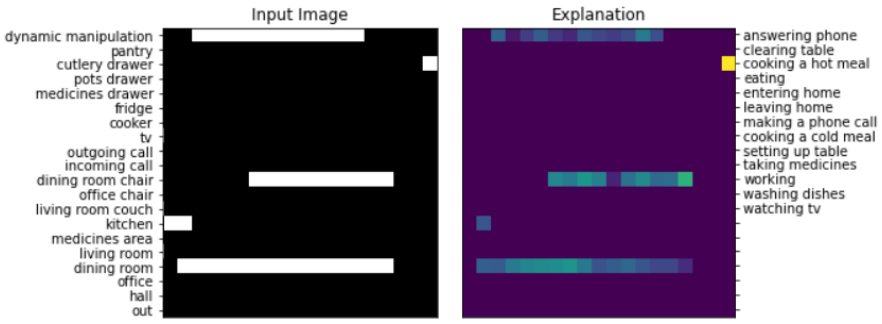


Figure 6.4: LIME: An example of an explanation for an input related to the *eating* activity. LIME deduced that the most important feature for classification was that the resident previously performed the activity *cooking a hot meal*

Model Prototypes

Finally, we considered the white-box deep explanation XAI method proposed in [130], which we will refer to as *Model Prototypes*. Differently from Grad-CAM and LIME that do not impose constraints on the specific DL model being used, *Model Prototypes* requires an autoencoder and a specifically designed neural network that is called *prototype classifier*. Given an input image img, the autoencoder reduces its dimensionality to derive low-level features that are effective for classification. The encoded image is provided as input to the *prototype classifier*. The *prototype classifier* is composed of three layers: a prototype layer, a fully-connected layer, and a softmax layer. During training, the *prototype classifier*

learns a set of p prototypes in the latent space. The prototypes are representatives of the training set data. During classification, the prototype layer computes the proximity in the latent space of the input image with the prototypes (using the squared distance). Hence, classification relies on the distance between the input and the learned prototypes.

In the original version of *Model Prototypes*, an explanation for an input image img is generated by showing its top m prototypes in terms of minimal distance to img in the latent space. For the sake of visualization, the prototypes are translated from the latent space to the original feature space using the decoder.

In this work, in order to obtain explanations that are comparable to the ones generated by Grad-CAM and LIME, we specifically designed a novel algorithm to generate a heat map by combining the input image with its $m-$closest prototypes. This process is described in Algorithm 2. Intuitively, for each white pixel in the input, its intensity in the heat map depends on how many of the m-closest prototypes have a non-zero value in the same pixel. In order to exclude noisy pixels from the prototypes, we only consider the ones with an intensity higher than a threshold pt.

Figure 6.5 shows an example of an explanation generated by Model Prototypes on the example image depicted in Figure 6.2 that was classified as *eating*, as well as the closest prototypes used to generate the explanation. Note that some prototypes may only exhibit slight differences between them. This is due to the fact that each prototype encodes a frequent pattern of activity segments, and different frequent patterns may have small differences that are crucial for classification. For instance, considering Figure 6.5, prototypes 1, 2, and 5 are very similar. Prototype 1 represents segments of the *eating* activity with continuous dynamic manipulations (e.g., the resident is using fork and knife) while sitting in the dining room. Prototype 2 represents segments of the *eating* activity performed while sitting in the dining room, without dynamic manipulations (e.g., the resident is not using fork and knife but just chewing). Finally, prototype 5 represents segments of the *eating* activity performed with discontinuous dynamic manipulation (e.g., the resident is drinking) while sitting in the dining room.

Algorithm 2 Generating a heatmap from prototypes

Input: The input image img, the m closest prototypes $P = \{p_1, p_2, \ldots, p_m\}$ to img, a threshold pt
Output: A heatmap h
$h \leftarrow$ empty heat map
for each row i of img **do**
    for each column j of img **do**
        if $img[i, j] > 0$ **then**
            $\tau \leftarrow 0$
            for $p \in P$ **do**
                if $p[i, j] > pt$ **then**
                    $\tau \leftarrow \tau + 1$
                end if
            end for
            $h[i, j] \leftarrow \frac{\tau}{m}$
        end if
    end for
end for
return h

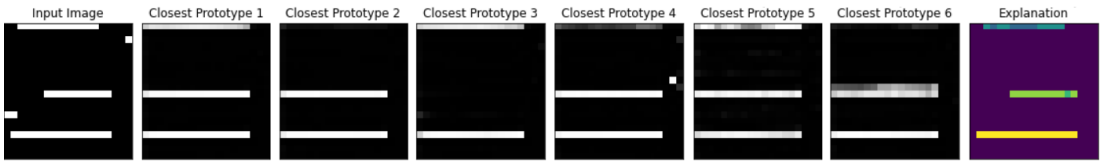


Figure 6.5: Model Prototypes: An example of an explanation for an input related to the *eating* activity, and the m-closest prototypes learned by the DL classifier. By comparing the prototypes and the input, the most important feature in the explanation is the presence of the resident in the dining room. Also, Model Prototypes deduced that sitting on the dining room chair was also relatively important

6.2.6 Generating semantic explanations in natural language

Each heat map generated by one of the XAI approaches described above can be analyzed to understand the rationale behind the corresponding classification instance. However, while these heat maps may be useful for data scientists, non-expert users (e.g., clinicians, caregivers, or the residents themselves) would

struggle to interpret them. Hence, we transform each heat map into a sentence in natural language. This process is divided into two steps: a) extracting the most relevant semantic features from the heat map, and b) using those semantic features to generate a sentence. In the following, we describe this process in detail.

Extracting relevant semantic features

From each heat map h generated as an explanation for an input image img classified as A, we extract a set of *relevant semantic features* $F^\star$, that is the union of the semantic states and the past activities that were relevant for the classification according to h. For each semantic feature $f \in F^\star$, we also compute its *relevance* $rel(f, img)$ based on the intensity of the corresponding pixels in h. In the following, we explain how we compute $F^\star$ and $rel(f, img)$.

Given a heat map h, we find the pixel $px^\star$ associated with the highest intensity. If the intensity level of $px^\star$ is over an *explainable threshold* et, we consider its belonging *segment of pixels* (i.e., the sequence of consecutive non-zero pixels on the same row where the pixel is located) as a relevant semantic feature f that we add to $F^\star$.[3] We assign to $rel(f, img)$ the intensity level of $px^\star$. We iterate this process until the intensity level of $px^\star$ (ignoring the ones belonging to *segments of pixels* that were previously included in $F^\star$) is lower than et. Note that the threshold et should be determined empirically for each method.

Generating the natural language sentence from the relevant semantic features

After we compute the relevance of each $f \in F^\star$ given an input image img, we generate a single natural language sentence including all the semantic features in $F^\star$ ordered by relevance. Note the relevance metric does not capture the temporal order of the features, but the importance that each feature has in the classification. We divide semantic features into four categories: *home objects usage*, *resident's position*, *low-level activities*, and *past activities*. We keep the

[3]Considering semantic states, a *segment of pixels* represents the occurrence of a specific type of semantic state within a time interval. The *segments of pixels* related to past activities only contain one pixel, and they indicate that a specific activity type is a past ADL.

semantic features of the same category close to each other in the sentence for the sake of user experience. We describe the temporal relationships between semantic features of the same category in natural language using temporal adverbs (e.g., *then*, *after*, *while*, ...). Algorithm 3 shows how this process works.

Algorithm 3 Generating a sentence in natural language

1: **Input:** An input image img, the classified activity A, the set of relevant semantic features $F^\star$, and $rel(f, img) \forall f \in F^\star$
2: **Output:** A sentence in natural language sen
3: $sen \leftarrow$ "*The activity is A*"
4: **while** $F^\star \neq \emptyset$ **do**
5: $\quad f^\star = argmax_{f \in F^\star} rel(f, img)$
6: $\quad F_C \leftarrow \{f \in F^\star | f$ is of the same category of $f^\star\}$
7: $\quad sen \leftarrow sen$ extended with a description of the semantic features in F_C ordered by $rel(f, img)$
8: $\quad F^\star \leftarrow F^\star \setminus F_C$
9: **end while**

In the following, for the sake of simplicity, we start showing examples of natural language explanations obtained by Algorithm 3 considering semantic features from the same category.

Example 6.5 *Consider the following home objects usage semantic features* $\{f_1 = Turned\_on\_cooker[1,3], f_2 = Pantry\_drawer\_opened[4,6]\}$*, where* $rel(f_1, img) = 0.8$ *and* $rel(f_2, img) = 0.6$*. DeXAR would generate the following description: Alice turned the cooker on and then opened the pantry.*

Example 6.6 *Consider the following resident's position semantic features* $\{f_1 = In\_the\_Office[2,5], f_2 = In\_the\_DiningRoom[7,9]\}$*, where* $rel(f_1, img) = 0.7$ *and* $rel(f_2, img) = 0.9$*. DeXAR would generate the following description: Bob has been in the dining room after being in the office.*

Example 6.7 *Consider the following low-level activities semantic features* $\{f_1 = Sitting[2,10], f_2 = Moving\_Arm[5,8]\}$*, where* $rel(f_1, img) = 0.84$ *and* $rel(f_2, img) = 0.74$*. DeXAR would generate the following description: Carl was sitting while moving his arm.*

Example 6.8 *Consider the following past activities semantic features* $\{f_1 = Performed\_Set\_up\_table[1], f_2 = Performed\_Cooking\_Hot\_Meal[2]\}$*, where* $rel(f_1, img) = 0.79$ *and* $rel(f_2, img) = 0.63$[4]*. DeXAR would generate the following description: Dave has just set up the table after cooking a hot meal.*

In the following, we show an application of Algorithm 3 considering semantic states from multiple categories.

Example 6.9 *Let Watching TV be the predicted activity, associated with a set of relevant semantic features* $F^\star = \{f_1 = Sitting[1, 12], f_2 = Television\_ON[4, 12]\}$*, where* $rel(f_1, img) = 0.82$ *and* $rel(f_2, img) = 0.78$*. DeXAR would generate the following description: The activity is Watching TV mainly because Eric was sitting while the television was on.*

Finally, we show how Algorithm 3 generates an explanation starting from the input image depicted in Figure 6.2 using *GradCAM*, *LIME*, and *Model Prototypes*.

Example 6.10 *Consider the image in Figure 6.2 that was classified as Eating. The application of our approach on the heat map in Figure 6.5 generated by Model Prototypes with* $et = 0.6$ *would result in the following sentence: The activity is Eating mainly because Bob has been in the dining room, and he was sitting on a chair of the dining room table. The same process applied on the heat map in Figure 6.3 generated by Grad-CAM with* $et = 0.8$ *would result in the following sentence: The activity is Eating mainly because Bob has been in the dining room, and he has just cooked a hot meal. Finally, the one obtained by LIME with* $et = 0.95$ *in Figure 6.4 would result in the following sentence: The activity is Eating mainly because Bob has just cooked a hot meal.*

6.3 Experimental evaluation

In this section, we describe our experimental evaluation to assess the quality of the explanations generated by DeXAR with the different XAI approaches. First,

[4]Note that the number associated with the past activities indicates their temporal order, from the most recent to the oldest.

we describe the datasets that we used in this research. Then, we explain the evaluation methodologies that we adopted to perform a quantitative evaluation. Finally, we show our main results.

6.3.1 Datasets

MARBLE

We first evaluate DeXAR on MARBLE [2], a dataset we collected in a controlled smart-home environment that we already introduced in Section 3.5.1. The smart-home environment was equipped with several environmental sensors: magnetic sensors on some doors and drawers (e.g., fridge, medicine drawer), pressure mat sensors on the chairs, and smart-plug sensors to detect the usage of home appliances (e.g., electrical cooker, TV). Each subject was carrying in the pocket a smartphone with an app in charge of detecting incoming and outgoing phone calls. The participants were also wearing a smartwatch in charge of collecting inertial sensor data (i.e., accelerometer, gyroscope, and magnetometer). The smart-home environment was divided into several semantic locations: *dining room*, *hall*, *kitchen*, *living room*, *medicine area*, and *office*. MARBLE includes data related to 13 ADLs: *answering phone*, *clearing table*, *cooking/cooking a hot meal*, *eating*, *getting in/entering Home*, *getting out/leaving home*, *making a phone call*, *preparing/cooking a cold meal*, *setting up table*, *taking medicines*, *working/using PC*, *washing dishes*, and *watching TV*. More details about the MARBLE dataset are described in Section 3.5.1.

While MARBLE includes inertial sensor data gathered from smartwatches, it does not contain annotations about low-level activities. In order to obtain such annotations, we decided to apply clustering methods on unlabeled inertial sensor data to infer meaningful arm manipulations that can be mapped to a semantic. After an accurate analysis based on PCA and K-Means (using the Silhouette score to compare different solutions), we derived two reliable clusters. The data points in the first cluster exhibit low variance, while the ones in the second cluster have higher variance. By correlating those clusters with ADLs, we observed that the first cluster was related to activities with limited arm movements (e.g., watching TV), while the second was related to activities with significant arm movements

(e.g., washing dishes). Then, we used a binary artificial neural network to classify *static* and *dynamic* manipulations using the labels derived from clustering. The classifier is a feed-forward fully connected network, with a dense layer of 64 neurons, and a softmax layer. Figure 6.6 shows the recognition rate using a leave-one-subject-out cross-validation.

	precision	recall	f1-score	support
STATIC MANIPULATION	0.9951	0.9945	0.9948	18005
DYNAMIC MANIPULATION	0.9905	0.9915	0.9910	10434
accuracy			0.9934	28439
macro avg	0.9928	0.9930	0.9929	28439
weighted avg	0.9934	0.9934	0.9934	28439

Figure 6.6: Recognition rate of the low-level activities classifier (leave-one-subject-out cross-validation)

The overall F1-score is around 0.99, hence our binary classifier is reliable in distinguishing static and dynamic manipulations. When a dynamic manipulation is classified from inertial sensors data, it is translated into a semantic state as explained in Section 6.2.1. For instance, a dynamic manipulation that occurred from $t1$ to $t2$ generates the semantic state $DynamicManipulation[t1, t2]$.

CASAS

We also evaluate DeXAR considering one of the CASAS datasets (i.e., the one named *Milan*) [171] since it has been extensively adopted for evaluation in the smart-home ADLs recognition literature [172]. The CASAS dataset includes annotated environmental sensor data collected in the home of a single resident. The smart home was mainly equipped with motion sensors that monitored the presence of the resident in the different home locations. The home was also equipped with two temperature sensors and a few magnetic sensors on doors and drawers. Unfortunately, CASAS does not contain inertial sensor data. For the sake of this work, we grouped ADLs as recently proposed in [172], excluding the ones that were poorly represented. Hence, we consider the following activities: *personal hygiene*, *dressing/undressing*, *kitchen activity*, *eating*, *watching TV*, *sleeping*, *reading*, *leaving home*, and *working*.

6.3.2 Evaluation methodologies

Evaluation based on common-sense knowledge

As a first assessment, we compared the different XAI approaches by evaluating the consistency of their explanations with respect to common-sense knowledge about the relationships between ADLs and semantic features[5]. The common-sense knowledge encodes high-level properties of the ADLs domain on which there is a general agreement by human beings. We typically acquire this knowledge during our life experiences. For instance, the *cooking* activity is commonly performed in the *kitchen*, interacting with cooking instruments like the *stove* and the *oven*. In the ADLs recognition literature, this knowledge has been often used in knowledge-based approaches [173].

In this chapter, knowledge is represented as a semantic model that defines, for each ADL, its *partially explaining semantic features*. A semantic feature f partially explains an activity A if f explains (even if partially) A according to common-sense knowledge. Our semantic model resembles a knowledge graph focused on a particular relationship.

For instance, the semantic state *fridge opened* partially explains both the *cooking a hot meal* and *taking medicines* activities, while it does not partially explain the activity *watching TV* even if it may actually occur while watching TV. We also model groups of semantic features that together partially explain activities. For instance, the semantic states *using kitchen chair* and *manipulating a fork* together partially explain the *eating* activity. Also, the past activities *cooking a cold meal* and *setting up table* partially explain the *eating* activity. Figure 6.7 depicts a small sample of our semantic model.

We quantitatively evaluate the quality of the automatically generated explanations according to our semantic model. Given A as the output of the classifier from an input image img, and $F^\star$ as the most relevant semantic features derived by an XAI approach, we compute the *common-sense relevance* $cr()$ for each semantic feature $f \in F^\star$:

[5]As we described in Section 6.2.6, a semantic feature is a semantic state or a past ADL.

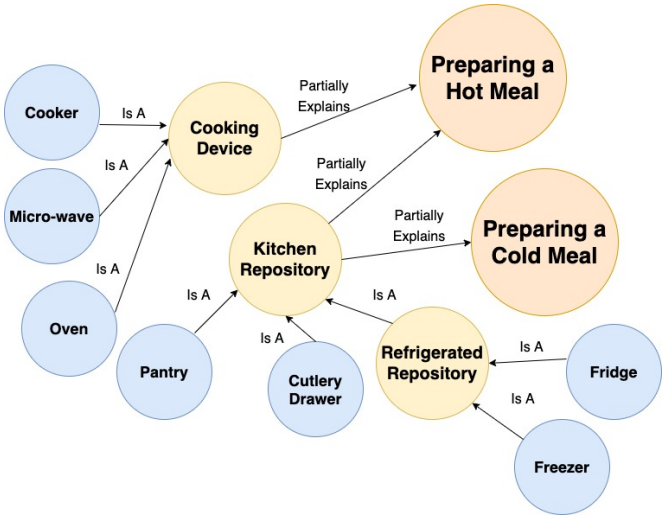


Figure 6.7: A small portion of our semantic model based on common-sense knowledge. Cooking devices (like the cooker, the microwave, and the oven) partially explain only the *preparing a hot meal* activity. On the other hand, kitchen repositories (that may also include the refrigerated ones), partially explain both *preparing a hot meal* and *preparing a cold meal* activities

$$cr(f, A) = \begin{cases} rel(f, img) & \textit{if } f \textit{ partially explains } A \\ -rel(f, img) & \textit{otherwise} \end{cases} \tag{6.5}$$

Hence, semantic features that do not partially explain the predicted activities are associated with a negative relevance, while the partially explaining ones are associated with a positive relevance. Based on the common-sense relevance, we compute the *Explanation Score* that takes values in the range $[-1, 1]$:

$$ExplanationScore(F^\star, A) = \begin{cases} \frac{\sum_{f \in F^\star} cr(f,A)}{\sum_{f \in F^\star} |cr(f,A)|} & \textit{if } F^\star \neq \emptyset \\ -1 & \textit{otherwise} \end{cases} \tag{6.6}$$

The common-sense relevance of each feature determines its impact on the final explanation score. If there are no semantic features in $F^\star$ that partially explain A, the explanation score is -1, while if every semantic feature partially explains A, the explanation score is 1.

User-based evaluation

In order to assess the effectiveness of explanations, we also wanted to understand how non-expert users would perceive them. First, we conducted a survey on the MARBLE dataset to obtain a user-based evaluation of the explanations generated by the three different approaches. Since we decided to evaluate DeXAR also on the CASAS dataset only after the first survey was completed, we conducted a second survey on CASAS a few months after the first one.

Overall, we recruited 84 subjects for the first survey (MARBLE dataset), and 63 for the second one (CASAS dataset). Participants were recruited through word-of-mouth, university mailing lists, and social media channels. These subjects have no experience in activity recognition and their age ranged from 20 to 60. In particular, the majority of participants ($\approx 78\%$) were younger adults with ages ranging from 18 to 30, while the remaining ones represented older subjects ($\approx 15\%$ with ages ranging from 31 to 50, and $\approx 7\%$ older than 50). The collected data are anonymous.

In order to achieve robust results, we generated different sets of system predictions randomly sampled from the test set. Each set includes a prediction for each activity with three explanations, each one obtained by one of the three considered methods. Each participant is first informed about the goal of our experiments, the home environment (with no mention about sensors and their positioning), and the considered activities. Then, the system randomly assigns to the participant one of the generated sets of system predictions. For the sake of this work, we only consider explanations associated with correct activity classifications and a high classifier's confidence. We asked the participants to vote for each explanation with a grade from 1 (absolutely not satisfying) to 5 (completely satisfying). The users were not aware of the method that generated each explanation. Figure 6.8 shows a partial screenshot of our survey.

6.3.3 Results

We performed a standard 70/10/20 partition of each dataset into training, validation, and test sets. For each XAI method, we produce explanations by providing the images generated from the test set as input to an XAI model trained on

Figure 6.8: A screenshot from our survey

images generated from the training set.

In the following, we show the values for each hyper-parameter of DeXAR, the recognition rates of our classification models based on a Convolutional Neural Network (CNN), and the assessment of the quality of our explanations both with the knowledge- and user-based evaluation methodologies.

Choice of Hyper-Parameters

In Section 6.2 we introduced several hyper-parameters. Some of those are related to the input pre-processing, while others are related to the generation of explanations. We performed a grid search to find the best parameters in order to optimize the overall F1 score as well as the knowledge-based explanation score. We observed some significant differences in the hyper-parameters considering the two datasets, as shown in Table 6.1.

In general, we observed that the most impacting factors are the average duration of ADLs and the number of participating users. Indeed, the segmentation window size n on MARBLE is 16 seconds, while it is 360 seconds for CASAS. This is due to the fact that CASAS considers a real deployment, while MARBLE was collected in controlled experiments where the activity duration was artificially reduced. Indeed, in MARBLE, the average duration of an activity instance is 50 seconds, while in CASAS is 30 minutes. Since longer segmentation windows lead to larger input images, the number of m-closest prototypes is higher in CASAS compared to MARBLE. This is due to the fact that *Model Prototypes* is based on a pixel-by-pixel match of the input with the prototypes, and larger images require more prototypes to generate reliable explanations. For the same reason,

Table 6.1: Hyperparameters in DeXAR

Hyperparameter	Description	MARBLE	CASAS
n	Length of semantic states segment	16s	360s
ov	Segmentation overlap	80%	80%
t	Consecutive predictions considered to derive past activities	3	2
c	Confidence threshold to derive past activities	0.75	0.60
K	Number of stable predictions considered in the input	2	2
p	Number of prototypes in Model Prototypes	500	100
pt	Threshold to exclude noisy pixels in prototypes	0.8	0.01
m	Number of prototypes closest to the input considered in Model Prototypes	6	33
l	Number of input perturbations created by LIME	1500	4000
et (LIME)	Explainability threshold for LIME	0.95	0.85
et (Grad-CAM)	Explainability threshold for Grad-CAM	0.8	0.03
et (Model Prototypes)	Explainability threshold for Model Prototypes	0.6	0.6

LIME required 4000 perturbations on CASAS compared to the 1000 required by MARBLE. Another significant difference between the two datasets is that MARBLE required 500 prototypes, while CASAS only 100. This is motivated by the fact that MARBLE involves 12 different subjects, each one with personal ADL patterns. On the other hand, CASAS includes data from only one subject performing repetitive patterns.

Hence, we believe that the type of the dataset (i.e., realistic vs controlled setting) and the number of involved subjects are strong indicators of the hyperparameters to choose when applying DeXAR in different domains.

Accuracy on ADL recognition

In the following, we describe the recognition rate obtained by our CNN models. We will refer as *CNN-GL* to the model used for Grad-CAM and LIME, and *CNN-MP* to the one used for Model Prototypes. We empirically determined the network structure of *CNN-GL*: a convolutional layer composed of 8 $2x2$ filters, followed by a flatten layer, and a softmax layer for classification. The simplicity

of this network is due to the fact that our images are small and the pixels are always positioned in well-defined positions based on their semantics. Hence, we experimentally observed that more complex models do not improve the recognition rate. Since *Model Prototypes* requires a specifically designed neural network to learn the prototypes, our *CNN-MP* is a slight adaptation of the network proposed in [130]. In particular, the minor changes are: 1) we reduced the number of convolutional layers both in the encoder and in the decoder (from 4 to 1) to extract features that are similar to the ones derived by *CNN-GL*, and 2) we empirically determined the number of generated prototypes. Figures 6.9 and 6.10 show the results of both models on both datasets.

	precision	recall	f1-score	support
Answering Phone	1.00	0.99	0.99	384
Clearing Table	0.41	0.51	0.45	104
Cooking a Cold Meal	0.69	0.72	0.70	168
Cooking a Hot Meal	0.87	0.79	0.83	331
Eating	0.97	1.00	0.98	505
Entering Home	0.98	0.91	0.94	113
Leaving Home	0.91	0.91	0.91	44
Making a Phone Call	0.99	0.99	0.99	158
Setting Up Table	0.64	0.51	0.56	180
Taking Medicines	0.25	0.92	0.39	25
Working	0.99	1.00	0.99	286
Washing Dishes	0.69	0.57	0.62	252
Watching TV	1.00	1.00	1.00	1066
accuracy			0.89	3616
macro avg	0.80	0.83	0.80	3616
weighted avg	0.90	0.89	0.90	3616

(a) CNN-GL

	precision	recall	f1-score	support
Answering Phone	0.99	1.00	0.99	384
Clearing Table	0.24	0.13	0.17	104
Cooking a Cold Meal	0.64	0.79	0.71	168
Cooking a Hot Meal	0.91	0.76	0.83	331
Eating	0.95	1.00	0.98	505
Entering Home	0.96	0.89	0.93	113
Leaving Home	0.89	0.73	0.80	44
Making a Phone Call	0.98	0.99	0.98	158
Setting Up Table	0.63	0.74	0.68	180
Taking Medicines	0.24	0.84	0.37	25
Working	0.99	0.99	0.99	286
Washing Dishes	0.82	0.67	0.73	252
Watching TV	1.00	1.00	1.00	1066
accuracy			0.90	3616
macro avg	0.79	0.81	0.78	3616
weighted avg	0.90	0.90	0.90	3616

(b) CNN-MP

Figure 6.9: MARBLE: Comparison of the recognition rates obtained by the different CNNs. *CNN-GL* is the model used for *Grad-CAM* and *LIME*, while *CNN-MP* is the one used for *Model Prototypes* with 500 learned prototypes.

We observed that both classifiers reach an overall weighted F1 of $\approx 90\%$ on MARBLE and $\approx 80\%$ on CASAS. The difference in recognition rate between the datasets is due to the fact that CASAS includes long-term data in a realistic deployment, hence the classification task is more difficult. Indeed, the annotations

	precision	recall	f1-score	support
Dressing/Undressing	0.54	0.20	0.30	226
Eating	0.93	0.20	0.33	71
Kitchen Activity	0.87	0.90	0.88	1003
Leaving Home	0.22	0.77	0.34	82
Personal Hygiene	0.80	0.74	0.77	408
Reading	0.88	0.85	0.87	933
Sleeping	0.88	0.91	0.90	2268
Watching TV	0.86	0.76	0.81	571
Working	0.92	0.89	0.90	185
accuracy			0.84	5747
macro avg	0.77	0.69	0.68	5747
weighted avg	0.85	0.84	0.83	5747

(a) CNN-GL

	precision	recall	f1-score	support
Dressing/Undressing	0.46	0.44	0.45	226
Eating	0.84	0.23	0.36	71
Kitchen Activity	0.80	0.87	0.83	1003
Leaving Home	0.14	0.16	0.15	82
Personal Hygiene	0.75	0.55	0.63	408
Reading	0.92	0.75	0.82	933
Sleeping	0.83	0.96	0.89	2268
Watching TV	0.81	0.61	0.70	571
Working	0.79	0.95	0.86	185
accuracy			0.80	5747
macro avg	0.70	0.61	0.63	5747
weighted avg	0.80	0.80	0.79	5747

(b) CNN-MP

Figure 6.10: CASAS: Comparison of the recognition rates obtained by the different CNNs. *CNN-GL* is the model used for *Grad-CAM* and *LIME*, while *CNN-MP* is the one used for *Model Prototypes* with 100 learned prototypes.

of CASAS are less accurate compared to MARBLE, and it also includes several noisy sensor measurements.

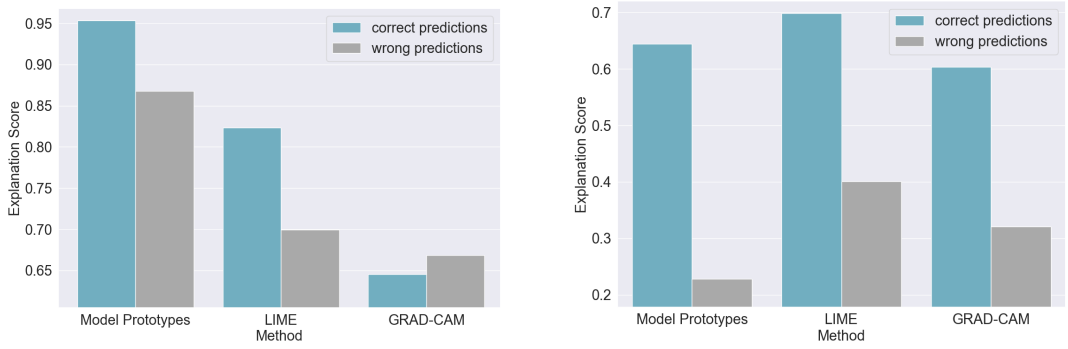
Considering both datasets, we observed a small difference in the F1 score between *CNN-MP* and *CNN-GL*. Hence, we consider these two models very similar since they are close in recognition rates, even if with minor differences in some activities. The similarity between *CNN-MP* and *CNN-GL* is crucial for a fair comparison of the different XAI techniques. In general, ADLs are reliably recognized, except for a few cases. For example, in MARBLE, the precision for *taking medicines* is low. This is due to the fact that this activity is performed in several locations of the home and it is poorly characterized by the deployed sensors (e.g., the resident can take medicines in the kitchen, taking water from the fridge, similarly to other kitchen-related activities). Other classification mistakes in MARBLE are related to setting up/clearing table activities. Those ADLs are monitored by a few sensors that also capture other kitchen-related activities. Those ADLs are often confused with each other due to their very similar patterns. Considering CASAS, the ADLs associated with the lowest recognition rates are *leaving home*, *eating*, and *dressing/undressing*. The low F1 score of *leaving home*

and *eating* is likely due to the fact that those activities are poorly represented in the dataset compared to the other ones. At the same time, both *eating* and *dressing/undressing* present a low recall value (i.e., they involve a high number of false negatives) since they are often confused with other activities that can be performed in the same locations of the considered smart-home (i.e., *kitchen activity* for *eating*, *reading* and *sleeping* for *dressing/undressing*).

We also compared these results with an approach based on a classic Deep feed-forward Neural Network (DNN) and raw sensor data, observing that it would be only $\approx 4\%$ better than *CNN-GL* in terms of overall macro F1 score. However, raw sensor data are poorly explainable. Hence, we sacrifice a bit of accuracy to take advantage of input data that encode semantics.

Explainability evaluation based on common-sense knowledge

In the following, we show the results of the evaluation based on common-sense knowledge. For the sake of visualization, we normalized the explanation score in the range $[0, 1]$. Figure 6.11 shows the explanation score reached by the different XAI approaches on both datasets. For each sample in the test set, we computed



(a) MARBLE (b) CASAS

Figure 6.11: Overall explanation score obtained by the different XAI approaches based on the common-sense knowledge evaluation

the explanation score using Equation 6.6. The figure shows the overall explanation score for correct and wrong classification instances separately. As expected, we observed a higher explanation score on correct predictions and a lower score

on wrong predictions. This is due to the fact that wrong predictions are often associated with semantic features that are not consistent with the predicted activity according to common-sense knowledge. The only exception is *Grad-CAM* on MARBLE, which reaches similar explanation scores both for correct and wrong predictions.

On MARBLE, we observed that *Model Prototypes* outperforms the other approaches. This is mainly due to the fact that the classifier is specifically designed to be explainable. Indeed, this method learns reliable prototypes that lead to good explanations. On the other hand, *LIME* and *Grad-CAM* aim to obtain explanations from existing CNNs. The bad performances of *Grad-CAM* are probably due to the fact that it often happens that the observed neurons are activated even if the corresponding features are not completely relevant for classification. This behavior of saliency-based approaches like *Grad-CAM* is well-known in the literature [129].

Considering CASAS, we observed a lower explanation score for all the XAI approaches. This is likely correlated with the lower recognition rate obtained by the underlying CNN-based models. On this dataset, *LIME* reaches a slightly higher explanation score compared to *Model Prototypes*. This is due to the fact that, since the input images consider a larger segmentation window, the generation of reliable prototypes is more challenging for those activities that are poorly represented. Indeed, looking closely at Figure 6.13, *LIME* is particularly better than *Model Prototypes* mainly for those activities that have few instances in the dataset (e.g., *eating* and *working*).

However, a significant advantage of *Model Prototypes* on this dataset is that it reaches a significantly low explanation score considering wrong predictions. Hence, when the classification is incorrect, the corresponding explanation would likely be unconvincing for the end-users, hence possibly pointing out to the users the miss-predictions of the system. On CASAS, *Grad-CAM* is still the approach associated with the lowest explanation score on correct predictions, even if not too distant from the other approaches.

Figures 6.12 and 6.13 show the overall explanation score (i.e. considering both correct and wrong predictions) at the ADL granularity on both datasets.

On MARBLE, *Model Prototypes* outperforms the other approaches on the

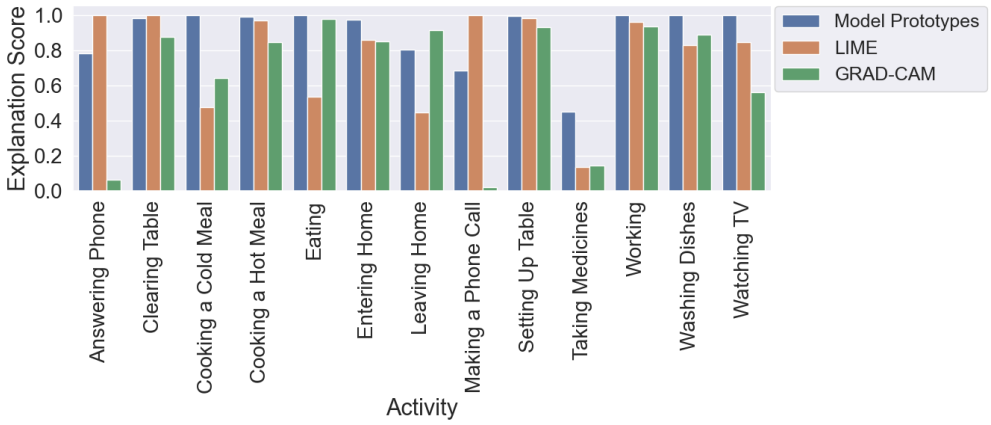


Figure 6.12: MARBLE: Explanation score for each activity obtained by the different XAI approaches based on the common-sense knowledge evaluation.

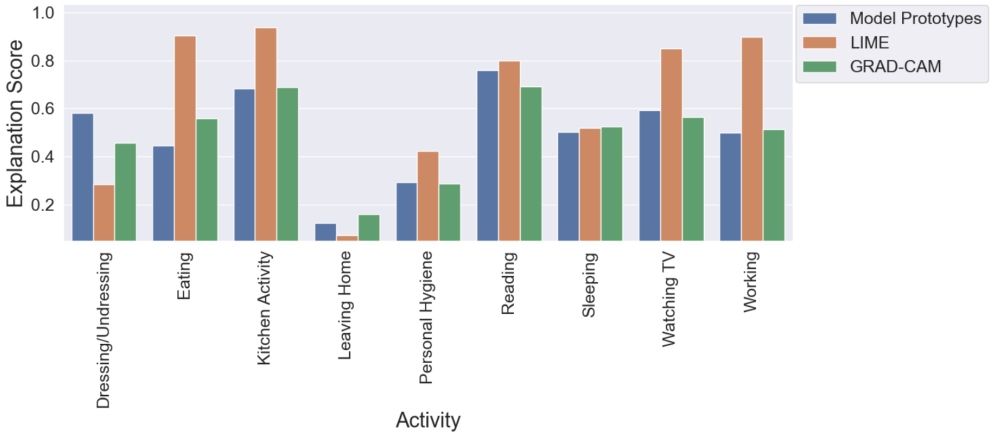


Figure 6.13: CASAS: Explanation score for each activity obtained by the different XAI approaches based on the common-sense knowledge evaluation.

majority of the activities. *LIME* performs better than *Model Prototypes* only on the phone-related activities. This is due to the fact that the only semantic features that partially explain those activities according to our common-sense knowledge are the ones related to phone usage. However, activities like *making a phone call* can be performed in every location of the home, standing or sitting. Hence, *Model Prototypes* generated noisy prototypes for this activity, while *LIME* better captures the important semantic features for classification. Due to the high variability of phone activities, *Grad-CAM* is not able to determine the important features only considering neuron activation, hence leading to

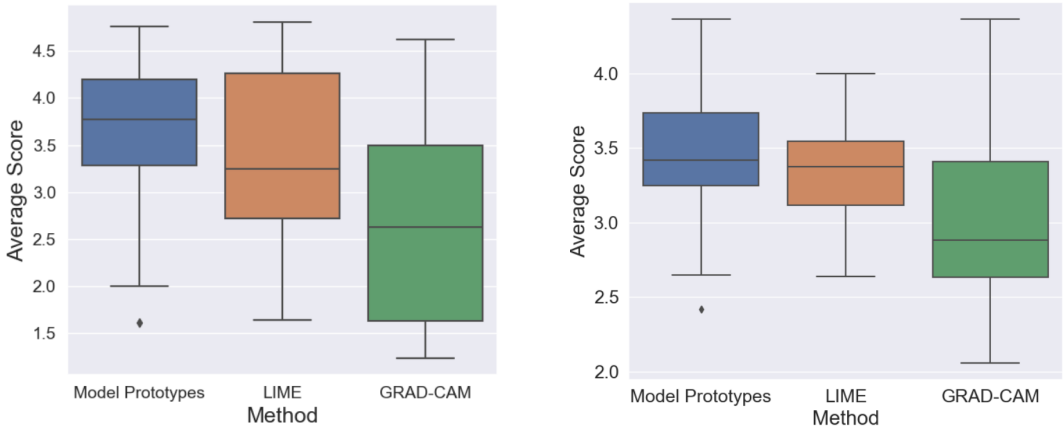
very poor explanation scores. Considering CASAS, *LIME* outperforms the other approaches regarding the activities that are poorly represented since the corresponding prototypes are less reliable. The higher explanation score on *kitchen activity* is due to the fact that (similarly to *making a phone call* on MARBLE) it can be performed in multiple home locations with different patterns. Indeed, *kitchen activity* includes different activities such as cooking, setting up and clearing the dining table, and drinking a glass of water in the kitchen. Hence, the prototypes related to this activity are less reliable.

An interesting insight is that poorly recognized activities are likely associated with bad explanations, independently from the XAI approach being used. This phenomenon occurs for *taking medicines* in MARBLE, and *leaving home* in CASAS. However, in MARBLE, *clearing table* has a high explanation score even if the recognition rate is low. This is due to the fact that this ADL is often confused with *setting up table*, and these activities share similar explanations (i.e., they involve the same sensors and the patterns are very similar).

Explainability evaluation based on the survey

In the following, we show the main results of our user-based survey. For the sake of fairness, we want to mention that we performed this evaluation only once the knowledge-based ones were completed. Hence, we did not adapt the semantic model based on data collected from the users. Figure 6.14 depicts the distribution of the average score for each explanation for each method on both datasets.

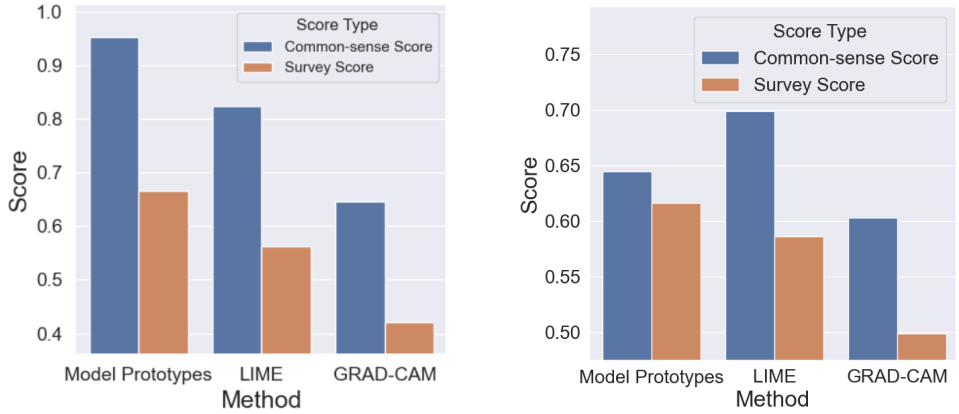
Considering both datasets, *Model Prototypes* is the most appreciated XAI approach by the participating users. On MARBLE, *Model Prototypes* has a higher average and lower variance compared to *LIME*. Considering CASAS, the difference is still clear since Model Prototypes statistically received higher rates. The results also confirm that *Grad-CAM* is the approach that generated significantly worse explanations, even from the participants' point of view. In order to summarize, Figure 6.15 directly compares the results of the survey (with explanation marks normalized in the interval $[0, 1]$) with the average explanation scores of the evaluation based on common-sense knowledge on both datasets. From the figure, it is interesting to notice how the evaluations based on the Explanation Score



(a) MARBLE (b) CASAS

Figure 6.14: Distribution of the scores provided by the participants for each method

(i.e., common-sense score) are overall proportional to the ones obtained through the surveys.



(a) MARBLE (b) CASAS

Figure 6.15: Comparison between common-sense knowledge evaluation and user-based evaluation

6.3.4 Impact of pre-processing hyper-parameters

In the following, we discuss how the main hyper-parameters that influence the generation of the input (i.e., the semantic images) impact the recognition rate on both datasets.

First, we show in Figure 6.16 how the length n of the segmentation window impacts the recognition rates. As we previously discussed, the optimal value

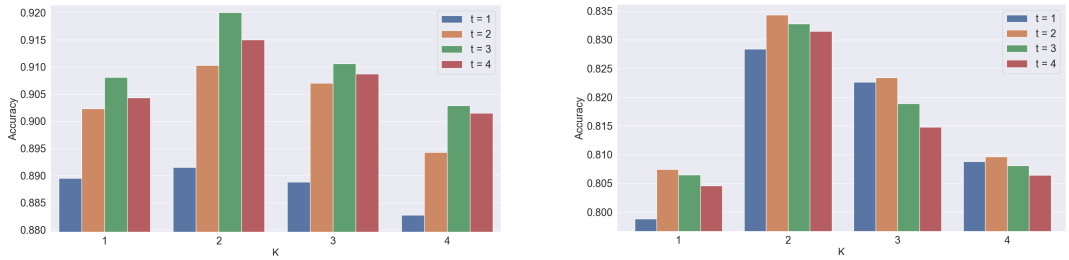
(a) MARBLE (b) CASAS

Figure 6.16: Impact of the segmentation window dimension (in seconds) on the recognition rates

of n is 16s for MARBLE and 360s for CASAS. Consistently with the activity recognition literature [3], we found that when n is lower than the best value, the input does not cover enough information to recognize ADLs. On the other hand, when n is higher than the best value, each window may consider sensor data from multiple activities, thus degrading the recognition rate.

Figure 6.17 shows the impact of the number of the past ADLs (K) and the number of consecutive predictions to consider past ADLs reliable (t). On both datasets, the optimal value of K is 2, while $t = 3$ in MARBLE and $t = 2$ in CASAS. We observed that high values of K negatively affect the recognition rate, since (as expected) only the most recently performed ADLs are informative to classify the current one (e.g., *cooking* and *eating* before *washing dishes*). Considering t, using low values leads to taking into account wrong classification instances as reliable past ADLs. On the other hand, high values introduce a delay in considering past ADLs in the input images, with a consequent slight negative impact on the recognition rate.

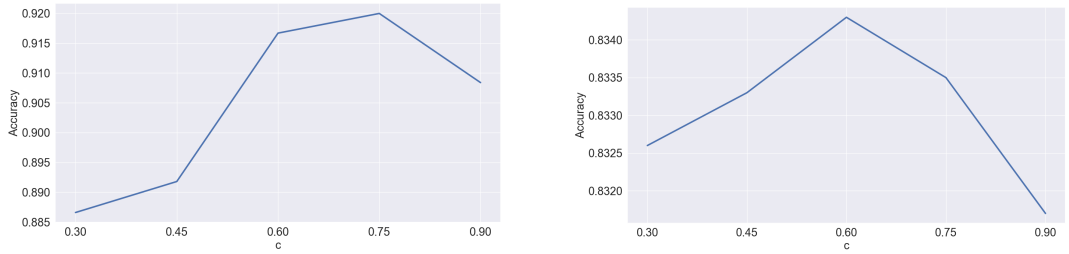
Figure 6.18 depicts how the confidence in past predictions (c) influences the



(a) MARBLE (b) CASAS

Figure 6.17: Impact of K (number of past activities) and t (consecutive reliable predictions) on the recognition rates

recognition rate. The optimal value of c is 0.75 on MARBLE and 0.60 for CASAS.



(a) MARBLE (b) CASAS

Figure 6.18: Impact of c (confidence threshold for past activities) on the recognition rates

Low values of c lead to consider a high number of miss-predictions as reliable past ADLs, with a negative impact on the recognition rate. On the other hand, high values significantly limit the number of considered past ADLs, negatively impacting the overall accuracy.

6.4 Discussion

In the following, we discuss the main limitations of DeXAR, pointing to possible future research directions to address them.

6.4.1 Over-reliance in explanations

The user study that we performed evaluates how explanations are rated by the users, mainly based on whether they were convincing or not. However, this type of evaluation may require extensions in higher-stakes domains, like ADL recognition for healthcare applications. Indeed, it is important not only to evaluate if an explanation is intuitive to the user but also if it is helpful for the specific application in a real-world scenario [174]. Explanations should provide information also to identify situations when the classifier is incorrect [175]. Otherwise, the end-users may wrongly build trust in the system.

In future work, we will investigate how to improve the user-based evaluation and how to mitigate the over-reliance problem. First, we will include additional information in the explanations. For instance, the classifier's confidence may help in indicating whether a specific ADL prediction is reliable or not. Another possibility is to link the explanation with the input of the classifier. However, differently from the image classification task, the input in our case are semantic images that are not easy to understand by non-expert users. Note that our images not only capture sensors' activations and semantic states but also their temporal relationships. Based on preliminary experiments, we observed that showing these images as the classifier input together with the explanation would only confuse the users. In future work, we will investigate how to represent the classifier's input in a user-friendly way, and how to include confidence. This task includes the design of specific user studies both for user-friendliness and for reliance.

6.4.2 Limitations of the *Model Prototypes* approach

In this work, we designed our own variant of the *Model Protoytpes* approach as presented in Section 6.2.5. While the achieved results are promising, this novel method still has some limitations that we plan to address in future work.

First, when considering the CASAS dataset, we observed that semantic states that happened to be active independently from the performed ADL (e.g., a drawer that is left open for a long time) are encoded in several prototypes, and, consequently, they appear as relevant features in many explanations. Clearly, this phenomenon has a negative impact on the quality of the explanations. We will

investigate alternative semantic image representations to mitigate this problem.

Another limitation is that each explanation is generated by computing a match pixel by pixel between the input and the m-closest prototypes. This approach may be too rigid. Indeed, a sequence of semantic states that occur at specific instants in the input may not be temporally aligned to the ones in the prototypes. We will investigate more sophisticated approaches considering temporal tolerance.

Finally, we generate an explanation by considering only semantic features that are both in the prototypes and in the input. However, prototypes may also include additional semantic features compared to the input. In future work, we will study how to include such information to enrich the explanations. This additional information may be useful to help the end-users in spotting classification mistakes, thus reducing the over-reliance problem described above. We illustrate this case with an example.

Example 6.11 *Alice is in the office, sitting at the desk while reading a book. However, the model wrongly classifies her current activity as working. The closest prototype, besides including the semantic states activated by Alice, also includes the semantic state Personal Computer ON that is not part of the input. This is due to the fact that, considering training data, Alice usually sits at the desk using the PC. The explanation produced by the system may use the additional semantic state in the prototype as follows: "The activity is Working mainly because Alice was sitting at the office's desk. However, the PC is off while this activity usually also includes using the PC".*

6.5 Summary

In this chapter, we presented DeXAR: a methodology for explainable sensor-based ADL recognition relying on the classification through deep learning of semantic images derived from raw sensor data. We quantitatively evaluated through the Explanation Score the effectiveness of the explanations generated by DeXAR. Our experiments showed how the results obtained through the Explanation Score are aligned with user-based scores obtained through surveys. Hence, our Explanation

Score addresses the research question **Q4** presented in Section 2.5 since it can be used to measure the degree of interpretability of DL models for sensor-based HAR.

However, due to time constraints, one of the main limitations of the work described in this chapter is that we used the Explanation Score only to evaluate purely data-driven approaches based on deep learning. As we will discuss in Chapter 7, we plan to exploit the Explanation Score to quantitatively evaluate the interpretability benefits provided by the Knowledge Infusion methods presented in this thesis, i.e., *symbolic features* and *semantic loss*.

Finally, another limitation of DeXAR is the transformation of raw sensor data into understandable semantic information. This transformation is crucial in order to take advantage of XAI methods and to evaluate their explanations through the Explanation Score. In essence, DeXAR relies on the conversion of raw sensor measurements into meaningful semantic states (e.g., hand gestures). This ensures that the generated explanations are comprehensible to end-users. For instance, when dealing with data from wearable devices, raw sensor measurements can be mapped to postures (e.g., the user is sitting) to produce better explanations about high-level activities (e.g., eating at the dining room table). However, this mapping can be challenging in certain scenarios. For instance, consider a scenario where a HAR application needs to recognize low-level physical activities like sitting, as discussed in chapters 4 and 5. In such cases, raw sensor measurements must be transformed into semantic states with an abstraction level that lies between the raw measurements and the physical activities. For instance, a repeated oscillatory hand movement can be derived from raw sensor data to recognize and explain activities like *running*. However, it remains an open question whether the end-users can easily understand this information when presented in an explanation. Consequently, when raw sensor data cannot be easily mapped to clear semantics, the Explanation Score can only partially evaluate the interpretability of a model: it can evaluate model interpretability according to the subset of input data encoding a clear semantic (e.g., the high-level context data in chapters 4 and 5). In the future, we plan to explore the use of more recent XAI methods specifically designed for multivariate time series and to extend our Explanation Score accordingly. In doing so, it will be also necessary to under-

stand how to translate explanations about raw sensor measurements into more accessible information that end-users can readily understand.

Chapter 7

Conclusion

7.1 Summary

In this thesis, we proposed novel Neuro-Symbolic AI (NeSy) methodologies to mitigate the labeled data scarcity issue of state-of-the-art approaches for sensor-based HAR. In particular, we introduced a NeSy framework for HAR in multi-subject smart environments and two NeSy approaches that implement the Knowledge Infusion paradigm for context-aware HAR. Moreover, we presented an initial investigation of interpretability aspects, by introducing a metric that quantitatively evaluates, according to domain knowledge, the explanations obtained from activity classifiers based on Deep Learning (DL). In the following, we summarize the specific contributions presented in this thesis.

Neuro-symbolic HAR in multi-subject smart-home environments
Our first contribution is the NeSy framework for HAR we presented in Chapter 3 to address data scarcity in multi-subject smart environments. This approach relies on symbolic reasoning to perform data association by combining sensor events with users' contextual information (e.g., their location in the environment). In this way, data association is performed without labeled data samples, thus mitigating data scarcity compared to existing state-of-the-art approaches in the field that perform data association in a data-driven fashion. More specifically, to further mitigate data scarcity, we implemented a semi-supervised DL classifier

that receives as input the streams of sensor events separated for each user after performing data association. This semi-supervised model was initialized with a limited amount of training data and then continuously updated and improved through a novel cache-based active learning strategy. Finally, symbolic reasoning is also considered to refine the classifier's predictions, by discarding those activities that are not consistent with the users' contextual information.

Our experimental evaluation on the MARBLE dataset revealed how the implemented classifier reliably recognized users' activities, without requiring labeled data to perform data association. In particular, the semi-supervised classifier reached similar recognition rates compared to a fully-supervised one, while requiring significantly lower labeled data and triggering a limited number of active learning queries. Moreover, prediction refinement improved the recognition rates of the activity classifier, while also reducing the number of active learning queries prompted to the user.

Among the limitations of this work, domain knowledge is exploited only before (i.e., to perform data association) and after (i.e., to perform prediction refinement) the learning process of the DL classifier. This does not allow the model to actually learn domain knowledge during training, thus limiting the potential benefits of NeSy solutions.

Knowledge infusion through symbolic features for context-aware HAR Labeled data scarcity affects also other HAR application domains, like the context-aware recognition of low-level (physical) activities. In this scenario, contextual information about the user's surroundings (e.g., her semantic location) can be used to better discriminate activities with similar motion patterns. NeSy approaches have already been considered to mitigate data scarcity in these applications that would require training sets containing every possible context condition in which activities can be performed. One of the main issues of existing NeSy approaches for context-aware HAR is that they consider domain knowledge about users' contextual information only after the training process of the DL classifier. This limits the ability of deep learning to handle data uncertainty and could hence lead to wrong decisions in case of incomplete knowledge models or noisy context data. For this reason, in Chapter 4 we presented a NeSy framework that

implements the Knowledge Infusion paradigm. Here, symbolic reasoning is used to infer additional knowledge-based features that are infused into the DL classifier. These features guide the model to learn domain knowledge in a less rigid way compared to state-of-the-art NeSy methods for context-aware HAR. More specifically, we implemented two versions of this NeSy approach. In the first case, symbolic reasoning relied on a standard ontology encoding hard domain constraints. In the second case, we instead considered a probabilistic ontology composed of both hard and soft constraints.

Our results on DOMINO and on a real-world dataset demonstrated that the use of symbolic features mitigates data scarcity while being more robust in the presence of noisy context data compared to more rigid NeSy approaches. Moreover, we showed how the improvements led by probabilistic ontologies do not justify the significant efforts required to build them.

Despite the promising results obtained thanks to the infusion of symbolic features into DL classifiers, a limitation of this approach is that such features must be inferred through computationally demanding symbolic reasoning procedures also during classification. This complicates the deployment of the proposed method on resource-constrained devices like smartphones.

Knowledge infusion through a semantic loss function for context-aware HAR

In Chapter 5 we address the limitation of the NeSy approach presented in Chapter 4, where symbolic reasoning is required also during classification, thus complicating the deployment of such a method on resource-constrained devices. Hence, we presented a Knowledge Infusion method based on a semantic loss function that infuses domain constraints into the DL classifier only during training, thus avoiding symbolic reasoning after deployment. In particular, this semantic loss penalizes those predictions that are not consistent with the users' surrounding context. In this way, after training, the DL model internally encodes domain knowledge that is exploited to classify context-consistent activities without requiring symbolic reasoning at run-time.

Our experiments analyzed the impact of different semantic loss functions that relied on both a standard and a probabilistic ontology. The results revealed how

our semantic loss approach outperformed a purely data-driven model. Moreover, it is the only NeSy method that can be deployed without the need for symbolic reasoning, reaching recognition rates that are close (or even better) to existing NeSy approaches. In addition, the use of a semantic loss is significantly more robust than the other considered NeSy approaches in the presence of noisy data. Finally, we inspect interpretability aspects, qualitatively showing how our semantic loss method makes decisions following the domain constraints encoded into the infused knowledge. This is a first step indicating how NeSy approaches based on Knowledge Infusion can lead to more interpretable DL classifiers.

One of the main problems of the work presented in Chapter 5 is that interpretability is analyzed briefly and only in a qualitative manner.

Explainable deep learning classifiers for sensor-based HAR

One of the potential benefits of NeSy solutions is to improve the interpretability of DL activity classifiers. However, eXplainable AI (XAI) methods for deep learning models are challenging to apply when input data are raw sensor measurements. Moreover, in the current HAR literature, no quantitative metric has been introduced to measure the interpretability level of DL models. Thus, in Chapter 6, we proposed a novel methodology to (i) transform raw sensor data in order to take advantage of existing XAI methods, (ii) use their output to generate explanations in natural language, and (iii) quantitatively evaluate such explanations through a novel metric called the Explanation Score. This metric measures the coherence of an explanation with human knowledge about the HAR domain.

Our experiments showed how evaluations based on the Explanation Score are aligned and consistent with user-based scores obtained through surveys. Hence, we believe that our metric can be used to quantify how DL models for sensor-based HAR are interpretable for humans.

However, due to time constraints, one of the main limitations of the framework presented in Chapter 6 is that we used the Explanation Score only to evaluate purely data-driven approaches based on deep learning. As we will discuss later, among the future research directions, we plan to exploit the Explanation Score also to evaluate the interpretability benefits provided by the NeSy methods described in this thesis.

7.2 Future work

Despite the encouraging results presented in this thesis, we believe that Neuro-Symbolic AI can further mitigate the labeled data scarcity and the lack of interpretability issues of sensor-based HAR. In the following, we outline some interesting and promising research directions that we plan to investigate in the future to improve our methods.

Including the explanation score into neuro-symbolic AI frameworks
In this thesis, we proposed the Explanation Score to quantitatively evaluate, based on domain knowledge, the consistency of explanations obtained through XAI methods from DL activity classifiers. However, we just exploited this score to evaluate the interpretability of purely data-driven classifiers. We believe that the Explanation Score can be integrated into NeSy frameworks for sensor-based HAR in several interesting ways.

To begin, this score can be used to evaluate if infusing domain knowledge into DL models during training increases their interpretability levels. This would confirm one of the main benefits of Neuro-Symbolic AI compared to approaches only based on deep learning, as also highlighted by the preliminary investigations on interpretability presented in Chapter 5.

In addition, in Chapter 6, we have seen how, when using specific XAI techniques (e.g., Model Prototypes), the Explanation Score tends to be higher for correct predictions and lower for wrong decisions. Hence, the metric can be potentially used to spot classification errors made by an activity classifier in real time when a relatively low Explanation Score is being computed.

Another interesting research direction consists of considering the Explanation Score to provide feedback to the DL model during its learning process in order to make it more interpretable. For instance, similarly to an existing approach proposed for monument facade image classification [176], the Explanation Score could be included in a custom training process that aligns the explanations of the DL model's predictions with the ones provided by human experts (encoded into a knowledge model). This would guide the activity classifier to intrinsically make predictions whose explanations are more interpretable for humans.

Finally, another limitation of the framework we proposed in Chapter 6 is that the Explanation Score can only be applied to high-level semantic data and not directly to raw sensor measurements. This leads to two possible alternative limitations. When both types of data are given as input to the DL classifier, the Explanation Score can only be used to evaluate the model's interpretability with respect to high-level semantic data. On the other hand, to avoid this problem, raw sensor data must be mapped to high-level semantic information before being provided as input to the classifier. In the first case, the Explanation Score can only be used to understand if the model is *partially* interpretable, i.e., it is interpretable only with respect to high-level semantic data. In the second case, mapping raw sensor measurements to a higher level of abstraction could lead to a loss of information (i.e., the fine-grained sensor patterns). In the future, it will be important to take into account more recent XAI methods specifically designed for multivariate time series. Hence, the Explanation Score should be extended to also consider these kinds of explanations.

Symbolic reasoning through Large Language Models

In chapters 3, 4 and 5, we introduced NeSy methods for HAR in multi-subject smart homes and for the context-aware recognition of low-level physical activities. These approaches include symbolic reasoning modules that rely on knowledge models (i.e., ontologies). However, building comprehensive and robust knowledge models (especially when probabilistic) is a challenging task that requires significant human effort and domain expertise. In the literature, some works already attempted to mitigate this problem by semi-automatically obtaining common sense knowledge from external sources (e.g., web [177], text sources [178], images [179]). However, these solutions mainly focused on the detection of users' activities in smart-home settings and the resulting models are still not sufficient for accurate recognition. In the future, we will investigate the possibility of replacing ontologies with Large Language Models (LLMs). Indeed, we believe that pre-trained LLMs inherently encode common sense knowledge about HAR (e.g., *walking* can be performed only with a positive speed). This knowledge can be hence infused into DL models. Proper experiments should be designed to evaluate (i) whether LLMs-based reasoning can serve as a good approximation of

ontological reasoning in HAR applications, (ii) whether the prompt engineering efforts required to build pipelines based on LLMs are more efficient compared to the intensive process required to build reliable ontologies, and (iii) whether solutions including LLMs are more flexible than the ones based on ontologies when, for example, new contextual information or additional activity classes need to be incorporated in the framework.

Neuro-symbolic self-supervised learning

In this thesis, we mainly focus on Neuro-Symbolic AI methods that mitigate the labeled data scarcity issue in the sensor-based HAR field. Self-Supervised Learning (SSL) is another technique that has been considered in the literature to tackle the same problem. This learning paradigm leverages large amounts of unlabeled data to pre-train a model capable of extracting reliable feature representation of sensor data. Hence, this pre-trained model is fine-tuned only using a limited amount of labeled data [113, 115]. Neuro-Symbolic AI can be potentially coupled with such techniques in different ways. For instance, symbolic reasoning can be involved during pre-training to make the model learn features that also take into account domain knowledge. At the same time, it could be possible to infuse domain constraints into the classifier during fine-tuning to further minimize the amounts of required labeled data.

Revising and updating the knowledge model through continual learning

In the Knowledge Infusion methods presented in chapters 4 and 5, we assumed that the knowledge model (i.e., the ontology) is static and never updated, even if this is not necessarily true in real-world deployments. Indeed, the knowledge model can be revised or extended with new knowledge over time. For instance, knowledge can be expanded by including new activities or context sources (in these cases, additional representative training samples are also required), or by refining and improving existing domain constraints. In this scenario, the presented NeSy methods based on Knowledge Infusion should re-train the DL model to infuse the updated knowledge. However, re-training the model from scratch is not always feasible (e.g., it is too expensive). Hence, continual learning ap-

proaches (e.g., based on knowledge distillation techniques) [180] could be adopted to incrementally update the classifier so that it can retain previous knowledge while learning new constraints. Applying existing continual learning approaches is presumably effective when the knowledge model is extended (e.g., with new activities), while it is more challenging when knowledge is revised. Indeed, in this case, the incremental learning paradigm should allow the model to retain some constraints while updating other ones. In the future, we plan to in-depth investigate how to incrementally train NeSy approaches upon changes in the knowledge model.

Bibliography

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

[2] L. Arrotta, C. Bettini, and G. Civitarese, "The marble dataset: Multi-inhabitant activities of daily living combining wearable and environmental sensors data," in *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pp. 451–468, Springer, 2021.

[3] 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.

[4] 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.

[5] 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.

[6] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, "Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–40, 2021.

[7] 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.

[8] 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.

[9] L. Chen, C. D. Nugent, L. Chen, and C. D. Nugent, "Sensor-based activity recognition review," *Human Activity Recognition and Behaviour Analysis: For Cyber-Physical Systems in Smart Environments*, pp. 23–47, 2019.

[10] 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.

[11] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *arXiv preprint arXiv:1708.08296*, 2017.

[12] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, "Neuro-symbolic artificial intelligence," *AI Communications*, vol. 34, no. 3, pp. 197–209, 2021.

[13] A. Sheth, M. Gaur, K. Roy, R. Venkataraman, and V. Khandelwal, "Process knowledge-infused ai: Toward user-level explainability, interpretability, and safety," *IEEE Internet Computing*, vol. 26, no. 5, pp. 76–84, 2022.

[14] K. Henricksen, J. Indulska, T. McFadden, and S. Balasubramaniam, "Middleware for distributed context-aware systems," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 846–863, Springer, 2005.

[15] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE pervasive computing*, vol. 16, no. 4, pp. 62–74, 2017.

[16] A. Benmansour, A. Bouchachia, and M. Feham, "Multioccupant activity recognition in pervasive smart home environments," *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, p. 34, 2016.

[17] S. N. Tran, D. Nguyen, T.-S. Ngo, X.-S. Vu, L. Hoang, Q. Zhang, and M. Karunanithi, "On multi-resident activity recognition in ambient smart-homes," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 3929–3945, 2020.

[18] T. Wang and D. J. Cook, "smrt: Multi-resident tracking in smart homes with sensor vectorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2809–2821, 2020.

[19] A. S. Crandall and D. Cook, "Attributing events to individuals in multi-inhabitant environments," 2008.

[20] L. Arrotta, C. Bettini, G. Civitarese, and R. Presotto, "Context-aware data association for multi-inhabitant sensor-based activity recognition," in *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pp. 125–130, IEEE, 2020.

[21] L. Arrotta, C. Bettini, and G. Civitarese, "Micar: Multi-inhabitant context-aware activity recognition in home environments," *Distributed and Parallel Databases*, pp. 1–32, 2022.

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

[23] 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.

[24] L. Arrotta, G. Civitarese, R. Presotto, and C. Bettini, "Domino: A dataset for context-aware human activity recognition using mobile devices," in *2023 24th IEEE International Conference on Mobile Data Management (MDM)*, pp. 346–351, IEEE, 2023.

[25] C. Bobed, R. Yus, F. Bobillo, and E. Mena, "Semantic reasoning on mobile devices: Do androids dream of efficient reasoners?," *Journal of Web Semantics*, vol. 35, pp. 167–183, 2015.

[26] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition—a survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1254, 2018.

[27] L. M. Dang, K. Min, H. Wang, M. J. Piran, C. H. Lee, and H. Moon, "Sensor-based and vision-based human activity recognition: A comprehensive survey," *Pattern Recognition*, vol. 108, p. 107561, 2020.

[28] P. Pareek and A. Thakkar, "A survey on video-based human action recognition: recent updates, datasets, challenges, and applications," *Artificial Intelligence Review*, vol. 54, pp. 2259–2322, 2021.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and mobile computing*, vol. 10, pp. 138–154, 2014.

[39] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880*, 2016.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

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

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] K. Gayathri, K. Easwarakumar, and S. Elias, "Probabilistic ontology based activity recognition in smart homes using markov logic network," *Knowledge-Based Systems*, vol. 121, pp. 173–184, 2017.

[52] M. Gaur, K. Faldu, and A. Sheth, "Semantics of the black-box: Can knowledge graphs help make deep learning systems more interpretable and explainable?," *IEEE Internet Computing*, vol. 25, no. 1, pp. 51–59, 2021.

[53] A. Sheth, M. Gaur, U. Kursuncu, and R. Wickramarachchi, "Shades of knowledge-infused learning for enhancing deep learning," *IEEE Internet Computing*, vol. 23, no. 6, pp. 54–63, 2019.

[54] U. Kursuncu, M. Gaur, and A. Sheth, "Knowledge infused learning (k-il): Towards deep incorporation of knowledge in deep learning," *arXiv preprint arXiv:1912.00512*, 2019.

[55] T. Dash, S. Chitlangia, A. Ahuja, and A. Srinivasan, "A review of some techniques for inclusion of domain-knowledge into deep neural networks," *Scientific Reports*, vol. 12, no. 1, p. 1040, 2022.

[56] J. Xu, Z. Zhang, T. Friedman, Y. Liang, and G. Broeck, "A semantic loss function for deep learning with symbolic knowledge," in *International conference on machine learning*, pp. 5502–5511, PMLR, 2018.

[57] K. Annervaz, P. K. Nath, and A. Dukkipati, "Actknow: Active external knowledge infusion learning for question answering in low data regime," *arXiv preprint arXiv:2112.09423*, 2021.

[58] R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, G. Cao, D. Jiang, M. Zhou, *et al.*, "K-adapter: Infusing knowledge into pre-trained models with adapters," *arXiv preprint arXiv:2002.01808*, 2020.

[59] K. Ahmed, S. Teso, K.-W. Chang, G. Van den Broeck, and A. Vergari, "Semantic probabilistic layers for neuro-symbolic learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 29944–29959, 2022.

[60] M. Fischer, M. Balunovic, D. Drachsler-Cohen, T. Gehr, C. Zhang, and M. Vechev, "Dl2: training and querying neural networks with logic," in *International Conference on Machine Learning*, pp. 1931–1941, PMLR, 2019.

[61] E. Giunchiglia and T. Lukasiewicz, "Coherent hierarchical multi-label classification networks," *Advances in neural information processing systems*, vol. 33, pp. 9662–9673, 2020.

[62] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing, "Harnessing deep neural networks with logic rules," *arXiv preprint arXiv:1603.06318*, 2016.

[63] A. S. A. Sukor, A. Zakaria, N. A. Rahim, L. M. Kamarudin, R. Setchi, and H. Nishizaki, "A hybrid approach of knowledge-driven and data-driven reasoning for activity recognition in smart homes," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 4177–4188, 2019.

[64] 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.

[65] W. Van Woensel, P. C. Roy, S. S. R. Abidi, and S. R. Abidi, "Indoor location identification of patients for directing virtual care: An ai approach using machine learning and knowledge-based methods," *Artificial Intelligence in Medicine*, vol. 108, p. 101931, 2020.

[66] G. Azkune and A. Almeida, "A scalable hybrid activity recognition approach for intelligent environments," *IEEE Access*, vol. 6, pp. 41745–41759, 2018.

[67] T. Xing, L. Garcia, M. R. Vilamala, F. Cerutti, L. Kaplan, A. Preece, and M. Srivastava, "Neuroplex: learning to detect complex events in sensor networks through knowledge injection," in *Proceedings of the 18th conference on embedded networked sensor systems*, pp. 489–502, 2020.

[68] J. Ye, G. Stevenson, and S. Dobson, "KCAR: A knowledge-driven approach for concurrent activity recognition," *Pervasive and Mobile Computing*, vol. 19, no. Supplement C, pp. 47–70, 2015.

[69] T. A. Nguyen, A. Raspitzu, and M. Aiello, "Ontology-based office activity recognition with applications for energy savings," *Journal of Ambient Intelligence and Humanized Computing*, vol. 5, pp. 667–681, 2014.

[70] A. Natani, A. Sharma, and T. Perumal, "Sequential neural networks for multi-resident activity recognition in ambient sensing smart homes," *Applied Intelligence*, vol. 51, pp. 6014–6028, 2021.

[71] H. Alemdar and C. Ersoy, "Multi-resident activity tracking and recognition in smart environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 4, pp. 513–529, 2017.

[72] A. Benmansour, A. Bouchachia, and M. Feham, "Modeling interaction in multi-resident activities," *Neurocomputing*, vol. 230, pp. 133–142, 2017.

[73] S. N. Tran, T.-S. Ngo, Q. Zhang, and M. Karunanithi, "Mixed-dependency models for multi-resident activity recognition in smart homes," *Multimedia Tools and Applications*, vol. 79, pp. 23445–23460, 2020.

[74] R. Mohamed, T. Perumal, M. N. Sulaiman, and N. Mustapha, "Multi resident complex activity recognition in smart home: A literature review," *Int. J. Smart Home*, vol. 11, no. 6, pp. 21–32, 2017.

[75] A. Lentzas, E. Dalagdi, and D. Vrakas, "Multilabel classification methods for human activity recognition: A comparison of algorithms," *Sensors*, vol. 22, no. 6, p. 2353, 2022.

[76] A. Alhamoud, V. Muradi, D. Böhnstedt, and R. Steinmetz, "Activity recognition in multi-user environments using techniques of multi-label classification," in *Proceedings of the 6th International Conference on the Internet of Things*, pp. 15–23, 2016.

[77] M. Jethanandani, T. Perumal, Y.-C. Liaw, J.-R. Chang, A. Sharma, and Y. Bao, "Binary relevance model for activity recognition in home environment using ambient sensors," in *2019 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pp. 1–2, IEEE, 2019.

[78] M. Jethanandani, A. Sharma, T. Perumal, and J.-R. Chang, "Multi-label classification based ensemble learning for human activity recognition in smart home," *Internet of Things*, vol. 12, p. 100324, 2020.

[79] R. Chen and Y. Tong, "A two-stage method for solving multi-resident activity recognition in smart environments," *Entropy*, vol. 16, no. 4, pp. 2184–2203, 2014.

[80] R. Mohamed, T. Perumal, M. N. Sulaiman, N. Mustapha, and M. Zainudin, "Modeling activity recognition of multi resident using label combination of multi label classification in smart home," in *AIP conference proceedings*, vol. 1891, AIP Publishing, 2017.

[81] R. Mohamed, M. N. S. Zainudin, T. Perumal, and S. Muhammad, "Adaptive profiling model for multiple residents activity recognition analysis using spatio-temporal information in smart home," in *Proceedings of the 8th International Conference on Computational Science and Technology: ICCST 2021, Labuan, Malaysia, 28–29 August*, pp. 789–802, Springer, 2022.

[82] N. Oukrich, A. Cherraqi, and D. Elghanami, "Multi-resident activity recognition method based in deep belief network," *J. Artif. Intell*, vol. 11, no. 2, pp. 71–78, 2018.

[83] T.-H. Tan, M. Gochoo, S.-C. Huang, Y.-H. Liu, S.-H. Liu, and Y.-F. Huang, "Multi-resident activity recognition in a smart home using rgb activity image and dcnn," *IEEE Sensors Journal*, vol. 18, no. 23, pp. 9718–9727, 2018.

[84] D. Chen, S. Yongchareon, E. M.-K. Lai, Q. Z. Sheng, and V. Liesaputra, "Locally weighted ensemble-detection-based adaptive random forest classifier for sensor-based online activity recognition for multiple residents," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13077–13085, 2021.

[85] D. Chen, S. Yongchareon, E. M.-K. Lai, J. Yu, and Q. Z. Sheng, "Hybrid fuzzy c-means cpd-based segmentation for improving sensor-based multiresident activity recognition," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11193–11207, 2021.

[86] D. Chen, S. Yongchareon, E. M.-K. Lai, J. Yu, Q. Z. Sheng, and Y. Li, "Transformer with bidirectional gru for nonintrusive, sensor-based activity recognition in a multiresident environment," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23716–23727, 2022.

[87] R. Kumar, I. Qamar, J. S. Virdi, and N. C. Krishnan, "Multi-label learning for activity recognition," in *2015 International Conference on Intelligent Environments*, pp. 152–155, IEEE, 2015.

[88] A. S. Crandall and D. J. Cook, "Using a hidden markov model for resident identification," in *2010 Sixth International Conference on Intelligent Environments*, pp. 74–79, IEEE, 2010.

[89] D. H. Wilson and C. Atkeson, "Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors," in *International Conference on Pervasive Computing*, pp. 62–79, Springer, 2005.

[90] C.-H. Lu and Y.-T. Chiang, "Interaction-feature enhanced multiuser model learning for a home environment using ambient sensors," *International journal of intelligent systems*, vol. 29, no. 11, pp. 1015–1046, 2014.

[91] 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.

[92] 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.

[93] 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.

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] 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.

[102] 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.

[103] 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.

[104] 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.

[105] 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.

[106] 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.

[107] 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.

[108] 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.

[109] 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.

[110] D. Riboni and F. Murru, "Unsupervised recognition of multi-resident activities in smart-homes," *IEEE Access*, vol. 8, pp. 201985–201994, 2020.

[111] J. Guo, Y. Li, M. Hou, S. Han, and J. Ren, "Recognition of daily activities of two residents in a smart home based on time clustering," *Sensors*, vol. 20, no. 5, p. 1457, 2020.

[112] T. Wang and D. J. Cook, "Multi-person activity recognition in continuously monitored smart homes," *IEEE transactions on emerging topics in computing*, vol. 10, no. 2, pp. 1130–1141, 2021.

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

[114] S. K. Hiremath, Y. Nishimura, S. Chernova, and T. Plötz, "Bootstrapping human activity recognition systems for smart homes from scratch," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 3, pp. 1–27, 2022.

[115] Y. Jain, C. I. Tang, C. Min, F. Kawsar, and A. Mathur, "Collossl: Collaborative self-supervised learning for human activity recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 1, pp. 1–28, 2022.

[116] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52138–52160, 2018.

[117] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information fusion*, vol. 58, pp. 82–115, 2020.

[118] C. T. Wolf, "Explainability scenarios: towards scenario-based xai design," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pp. 252–257, 2019.

[119] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and R. Helaoui, "Smartfaber: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment," *Artificial intelligence in medicine*, vol. 67, pp. 57–74, 2016.

[120] G. D. A. DW, "Darpa's explainable artificial intelligence program," *AI Mag*, vol. 40, no. 2, p. 44, 2019.

[121] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model," 2015.

[122] N. Narodytska, A. Ignatiev, F. Pereira, J. Marques-Silva, and I. Ras, "Learning optimal decision trees with sat.," in *Ijcai*, pp. 1362–1368, 2018.

[123] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[124] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[125] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[126] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature machine intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[127] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: interpreting, explaining and visualizing deep learning*, vol. 11700. Springer Nature, 2019.

[128] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[129] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, "The (un) reliability of saliency methods," *Explainable AI: Interpreting, explaining and visualizing deep learning*, pp. 267–280, 2019.

[130] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[131] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, "This looks like that: deep learning for interpretable image recognition," *Advances in neural information processing systems*, vol. 32, 2019.

[132] S. Mohseni, N. Zarei, and E. D. Ragan, "A multidisciplinary survey and framework for design and evaluation of explainable ai systems," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 11, no. 3-4, pp. 1–45, 2021.

[133] T. Kulesza, S. Stumpf, M. Burnett, W.-K. Wong, Y. Riche, T. Moore, I. Oberst, A. Shinsel, and K. McIntosh, "Explanatory debugging: Supporting end-user debugging of machine-learned programs," in *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 41–48, IEEE, 2010.

[134] M. Bilgic and R. J. Mooney, "Explaining recommendations: Satisfaction vs. promotion," in *Beyond personalization workshop, IUI*, vol. 5, p. 153, 2005.

[135] M. Nourani, S. Kabir, S. Mohseni, and E. D. Ragan, "The effects of meaningful and meaningless explanations on trust and perceived system accuracy in intelligent systems," in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 7, pp. 97–105, 2019.

[136] B. Herman, "The promise and peril of human evaluation for model interpretability," *arXiv preprint arXiv:1711.07414*, 2017.

[137] C. Roy, M. Shanbhag, M. Nourani, T. Rahman, S. Kabir, V. Gogate, N. Ruozzi, and E. D. Ragan, "Explainable activity recognition in videos.," in *IUI Workshops*, vol. 2, 2019.

[138] S. Suzuki, Y. Amemiya, and M. Sato, "Skeleton-based explainable human activity recognition for child gross-motor assessment," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pp. 4015–4022, IEEE, 2020.

[139] T. Zhuo, Z. Cheng, P. Zhang, Y. Wong, and M. Kankanhalli, "Explainable video action reasoning via prior knowledge and state transitions," in *Proceedings of the 27th acm international conference on multimedia*, pp. 521–529, 2019.

[140] C. Bettini, G. Civitarese, and M. Fiori, "Explainable activity recognition over interpretable models," in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 32–37, IEEE, 2021.

[141] M. Atzmueller, N. Hayat, M. Trojahn, and D. Kroll, "Explicative human activity recognition using adaptive association rule-based classification," in *2018 IEEE International Conference on Future IoT Technologies (Future IoT)*, pp. 1–6, IEEE, 2018.

[142] H. W. Guesgen, "Using rough sets to improve activity recognition based on sensor data," *Sensors*, vol. 20, no. 6, p. 1779, 2020.

[143] E. Khodabandehloo, D. Riboni, and A. Alimohammadi, "Healthxai: Collaborative and explainable ai for supporting early diagnosis of cognitive decline," *Future Generation Computer Systems*, vol. 116, pp. 168–189, 2021.

[144] Q. Li, R. Gravina, Y. Li, S. H. Alsamhi, F. Sun, and G. Fortino, "Multi-user activity recognition: Challenges and opportunities," *Information Fusion*, vol. 63, pp. 121–135, 2020.

[145] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[146] L. St, S. Wold, *et al.*, "Analysis of variance (anova)," *Chemometrics and intelligent laboratory systems*, vol. 6, no. 4, pp. 259–272, 1989.

[147] B. Settles, "Active learning literature survey," tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 2009.

[148] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and V. Bulgari, "From lab to life: Fine-grained behavior monitoring in the elderly's home," in *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 342–347, IEEE, 2015.

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

[150] R. Zhang, R. Xue, and L. Liu, "Searchable encryption for healthcare clouds: A survey," *IEEE Transactions on Services Computing*, vol. 11, no. 6, pp. 978–996, 2017.

[151] A. Ferrari, D. Micucci, M. Mobilio, and P. Napoletano, "On the personalization of classification models for human activity recognition," *IEEE Access*, vol. 8, pp. 32066–32079, 2020.

[152] 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.

[153] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Journal of Web Semantics*, vol. 5, no. 2, pp. 51–53, 2007.

[154] 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.

[155] 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.

[156] J. Noessner and M. Niepert, "Elog: a probabilistic reasoner for owl el," in *International Conference on Web Reasoning and Rule Systems*, pp. 281–286, Springer, 2011.

[157] F. Cruciani, A. Vafeiadis, C. Nugent, I. Cleland, P. McCullagh, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, and R. Hamzaoui, "Feature learning for human activity recognition using convolutional neural networks: A case study for inertial measurement unit and audio data," *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 1, pp. 18–32, 2020.

[158] P. Tarafdar and I. Bose, "Recognition of human activities for wellness management using a smartphone and a smartwatch: a boosting approach," *Decision Support Systems*, vol. 140, p. 113426, 2021.

[159] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, 2017.

[160] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[161] S. Ha, J.-M. Yun, and S. Choi, "Multi-modal convolutional neural networks for activity recognition," in *2015 IEEE International conference on systems, man, and cybernetics*, pp. 3017–3022, IEEE, 2015.

[162] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition.," in *Ijcai*, vol. 15, pp. 3995–4001, Buenos Aires, Argentina, 2015.

[163] B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, *et al.*, "Owl 2 web ontology language profiles," *W3C recommendation*, vol. 27, no. 61, 2009.

[164] P. Agarwal and M. Alam, "A lightweight deep learning model for human activity recognition on edge devices," *Procedia Computer Science*, vol. 167, pp. 2364–2373, 2020.

[165] 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.

[166] D. Wu, S.-J. Zheng, C.-A. Yuan, and D.-S. Huang, "A deep model with combined losses for person re-identification," *Cognitive Systems Research*, vol. 54, pp. 74–82, 2019.

[167] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," *arXiv preprint arXiv:1806.07421*, 2018.

[168] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognition*, vol. 41, no. 6, pp. 2010–2024, 2008.

[169] Z. Yan, D. Chakraborty, A. Misra, H. Jeung, and K. Aberer, "Sammple: Detecting semantic indoor activities in practical settings using locomotive signatures," in *2012 16th International Symposium on Wearable Computers*, pp. 37–40, Ieee, 2012.

[170] M. Gochoo, T.-H. Tan, S.-C. Huang, S.-H. Liu, and F. S. Alnajjar, "Dcnn-based elderly activity recognition using binary sensors," in *2017 international conference on electrical and computing technologies and applications (ICECTA)*, pp. 1–5, IEEE, 2017.

[171] D. J. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment," *Methods of information in medicine*, vol. 48, no. 05, pp. 480–485, 2009.

[172] D. Liciotti, M. Bernardini, L. Romeo, and E. Frontoni, "A sequential deep learning application for recognising human activities in smart homes," *Neurocomputing*, vol. 396, pp. 501–513, 2020.

[173] L. Chen, C. D. Nugent, and H. Wang, "A knowledge-driven approach to activity recognition in smart homes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 961–974, 2011.

[174] G. Bansal, T. Wu, J. Zhou, R. Fok, B. Nushi, E. Kamar, M. T. Ribeiro, and D. Weld, "Does the whole exceed its parts? the effect of ai explanations on complementary team performance," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2021.

[175] Z. Buçinca, M. B. Malaya, and K. Z. Gajos, "To trust or to think: cognitive forcing functions can reduce overreliance on ai in ai-assisted decision-making," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, pp. 1–21, 2021.

[176] N. Díaz-Rodríguez, A. Lamas, J. Sanchez, G. Franchi, I. Donadello, S. Tabik, D. Filliat, P. Cruz, R. Montes, and F. Herrera, "Explainable neural-symbolic learning (x-nesyl) methodology to fuse deep learning representations with expert knowledge graphs: The monumai cultural heritage use case," *Information Fusion*, vol. 79, pp. 58–83, 2022.

[177] 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.

[178] K. Y. Yordanova, "Texttohbm: A generalised approach to learning models of human behaviour for activity recognition from textual instructions.," in *AAAI Workshops*, 2017.

[179] D. Riboni and M. Murtas, "Web mining & computer vision: New partners for object-based activity recognition," in *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 158–163, IEEE, 2017.

[180] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, "Embracing change: Continual learning in deep neural networks," *Trends in cognitive sciences*, vol. 24, no. 12, pp. 1028–1040, 2020.