Recognizing and predicting activities in smart homes

Julien Cumin

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Julien CUMIN

Thèse dirigée par **James L. Crowley**, Professeur, Grenoble INP,
et codirigée par **Grégoire Lefebvre**, Chercheur, Orange Labs,
et codirigée par **Fano Ramparany**, Chercheur, Orange Labs.

préparée au sein du **Laboratoire d'Informatique de Grenoble**
dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Reconnaissance et prédiction d'activités dans la maison connectée

Recognizing and predicting activities in smart homes

Thèse soutenue publiquement le **4 décembre 2018**,
devant le jury composé de :

M. Patrick Brézillon
Professeur émérite, Sorbonne Université, Rapporteur
Mme Gaëlle Calvary
Professeur, Grenoble INP, Présidente du jury
M. James L. Crowley
Professeur, Grenoble INP, Directeur de thèse
M. Grégoire Lefebvre
Chercheur, Orange Labs, Co-encadrant de thèse
M. Fano Ramparany
Chercheur, Orange Labs, Co-encadrant de thèse
M. Daniel Roggen
Professeur associé, University of Sussex, Examinateur
M. Albrecht Schmidt
Professeur, Ludwig-Maximilians-Universität München, Rapporteur
M. Jean-Yves Tigli
Maitre de conférences, Université de Nice Sophia-Antipolis, Examinateur

Abstract

Understanding the context of a home is essential in order to provide services to occupants that fit their situations and thus fulfil their needs. One example of service that such a context-aware smart home could provide is that of a communication assistant, which can for example advise correspondents outside the home on the availability for communication of occupants. In order to implement such a service, it is indeed required that the home understands the situations of occupants, in order to derive their availability.

In this thesis, we first propose a definition of context in homes. We argue that one of the primary context dimensions necessary for a system to be context-aware is the activity of occupants. As such, we then study the problem of recognizing activities, from ambient smart home sensors. We propose a new supervised place-based approach which both improves activity recognition accuracy as well as computing times compared to standard approaches.

Smart home services, such as our communication assistance example, may often need to anticipate future situations. In particular, they need to anticipate future activities of occupants. Therefore, we design a new supervised activity prediction model, based on previous state-of-the-art work. We propose a number of extensions to improve prediction accuracy based on the specificities of smart home environments.

Finally, we study the problem of inferring the availability of occupants for communication, in order to illustrate the feasibility of our communication assistant example. We argue that availability can be inferred from primary context dimensions such as place and activity (which can be recognized or predicted using our previous contributions), and by taking into consideration the correspondent initiating the communication as well as the modality of communication used. We discuss the impact of the activity recognition step on availability inference.

We evaluate those contributions on various state-of-the-art datasets, as well as on a new dataset of activities and availabilities in homes which we constructed specifically for the purposes of this thesis: Orange4Home. Through our con-

tributions to these 3 problems, we demonstrate the way in which an example context-aware communication assistance service can be implemented, which can advise on future availability for communication of occupants. More generally, we show how secondary context dimensions such as availability can be inferred from other context dimensions, in particular from activity. Highly accurate activity recognition and prediction are thus mandatory for a smart home to achieve context awareness.

Keywords Smart Home, Internet of Things, Context Awareness, Activity Recognition, Activity Prediction, Availability Estimation, Machine Learning, Dynamic Bayesian Networks.

Résumé

Comprendre le contexte ambiant d'une maison est essentiel pour pouvoir proposer à ses occupants des services adaptés à leurs situations de vie, et qui répondent donc à leurs besoins. Un exemple de tel service est un assistant de communication, qui pourrait par exemple informer les personnes hors de la maison à propos de la disponibilité des habitants de celle-ci pour communiquer. Pour implémenter un tel service, il est en effet nécessaire que la maison prenne en compte les situations de ses occupants, pour ensuite en déduire leurs disponibilités.

Dans cette thèse, nous nous intéressons dans un premier temps à définir ce qu'est le contexte dans une maison. Nous défendons que l'activité des occupants est l'une des dimensions principales du contexte d'une maison, nécessaire à la mise en œuvre de systèmes sensibles au contexte. C'est pourquoi nous étudions dans un second temps le problème de la reconnaissance automatique d'activités humaines, à partir des données de capteurs ambiants installés dans la maison. Nous proposons une nouvelle approche d'apprentissage automatique supervisé basée sur les lieux de la maison, qui améliore à la fois les performances de reconnaissance correcte d'activités ainsi que les temps de calcul nécessaires, par rapport aux approches de l'état de l'art.

Par ailleurs, ces services sensibles au contexte auront probablement besoin de pouvoir anticiper les situations futures de la maison. En particulier, ils doivent pouvoir anticiper les activités futures réalisées par les occupants. C'est pourquoi nous proposons un nouveau modèle de prédiction supervisée d'activités, basé sur des modèles de l'état de l'art. Nous introduisons un certain nombre d'extensions à ce modèle afin d'améliorer les performances de prédiction, en se basant sur des spécificités des environnements de maisons instrumentées.

Enfin, nous nous intéressons à l'estimation de la disponibilité des occupants à communiquer, afin d'illustrer la faisabilité de notre exemple de service d'assistance à la communication. Nous suggérons que la disponibilité peut être inférée à partir des dimensions primaires du contexte, comme le lieu et l'activité (que l'on peut reconnaitre et prédire à l'aide de nos contributions précédentes), mais en

prenant également en compte le correspondant initiant la communication, ainsi que la modalité utilisée. Nous discutons de l'impact de l'étape de reconnaissance d'activités sur l'estimation de la disponibilité.

Nous évaluons expérimentalement ces contributions sur différents jeux de données de l'état de l'art, ainsi que sur un nouveau jeu de données d'activités et de disponibilités dans la maison que nous avons spécifiquement construit durant cette thèse : Orange4Home. À travers nos contributions à ces trois problèmes, nous démontrons l'implémentabilité d'un service d'assistance à la communication, pouvant conseiller des correspondants extérieurs sur les futures disponibilités des occupants de la maison. De manière plus générale, nous montrons comment des dimensions secondaires du contexte, comme la disponibilité, peuvent être inférées d'autres dimensions du contexte, comme l'activité. Il est donc essentiel pour qu'une maison devienne sensible au contexte, que celle-ci dispose de systèmes de reconnaissance et de prédiction d'activités les plus fiables possibles.

Mots-clés Maison intelligente, Internet des objets, Sensibilité au contexte, Reconnaissance d'activités, Prédiction d'activités, Estimation de disponibilité, Apprentissage automatique, Réseaux bayésiens dynamiques.

Remerciements

Je remercie les relecteurs Patrick Brézillon et Albrecht Schmidt, ainsi que les examinateurs Gaëlle Calvary, Daniel Roggen, et Jean-Yves Tigli pour l'intérêt qu'ils ont portés à mes travaux.

Je remercie mes trois encadrants, James L. Crowley, Grégoire Lefebvre, et Fano Ramparany, pour avoir partagé avec moi leurs immenses expériences de chercheurs, qui ont grandement contribuées à l'aboutissement de ces travaux. Leurs suggestions avisées, leurs encouragements, leurs critiques, et leurs amitiés m'ont permis de grandement progresser sur le plan scientifique, professionnel, et humain durant ces trois années de thèse.

Je remercie les membres du projet Amiqual4Home, et en particulier Nicolas Bonnefond et Stan Borkowski, pour m'avoir permis d'utiliser leurs ressources (et surtout leurs temps) afin de mener à bien mes expérimentations.

Je tiens ensuite à remercier tous mes collègues actuels et passés de l'équipe CASH (anciennement SMARTHOME (anciennement COSY) et COST), et plus généralement les personnes avec qui je suis allé manger des steaks-frites les midis. Je remercie également mes anciens collègues de l'équipe TAP (assassinée par une réorganisation), ainsi que les nombreux stagiaires que j'ai pu croiser. Merci en particulier à Samuel (qui m'a montré la voie), Loïc (pour les discussions sportives), Thomas (qui devrait se remettre aux rollers), et à tous mes collègues de bureau actuels et passés : Ravi (à qui j'ai succédé comme thésard de l'équipe), Kévin (pour les soirées jeux), Aimé (que je forcerai à sortir de la retraite pour venir à ma soutenance, bien qu'elle ne parle pas d'ellipsométrie), Samuli (qui n'aime pas la climatisation), Catherine (qui devrait arrêter de travailler le weekend), et David (dont la déconversion est en bonne voie).

Je remercie mes parents pour m'avoir permis d'arriver là où je suis, mes deux frères qui auront l'obligation de m'appeler « Dr. Cumin » jusqu'à la fin de leurs jours, et Dolly (que j'espère voir soutenir une thèse !).

Julien

Contents

Nomenclature

Abbreviations

AAL	Ambient Assisted Living. 14, 15, 22, 33, 84
ADL	Activity of Daily Living. 9, 13, 16, 34, 35, 39, 40
AI	Artificial Intelligence. 34, 37, 75
ANN	Artificial Neural Network. 38, 39, 52, 53
ATMS	Assumption-based Truth Maintenance System. 36
BN	Bayesian Network. 52, 55, 62, 64–66, 71, 73, 84, 88, 89, 99, 100
CEFA	CurrEnt Features and activity to predict the next Activity. 90, 98–100
CNN	Convolutional Neural Network. 39
CRAFFT	CuRrent Activity and Features to predict next FeaTures. 85–87, 89–108
CRF	Conditional Random Field. 83, 85, 86
CSCRAFFT	Cognitive State CRAFFT. 95, 96, 98, 104–107
DBN	Dynamic Bayesian Network. 77, 85–108
DST	Dempster-Shafer Theory. 57–59, 70
DTW	Dynamic Time Warping. 52, 55, 56, 64–66, 72, 73
DWRMSE	Duration-Weighted Root Mean Square Error. 121–127
HAR	Human Activity Recognition. 31–33, 37, 42
HCI	Human-Computer Interaction. 33, 95
HMM	Hidden Markov Model. 52, 83, 84

LSTM	Long Short-Term Memory neural network. 85, 101
MLP	MultiLayer Perceptron. 52, 53, 58, 62, 64–66, 69–74, 86, 90, 99, 100, 122, 125
NMCRAFFT	Non-Markovian CRAFFT. 93, 94, 96–98, 102–107
OWL	Web Ontology Language. 35
PDA	Personal Digital Assistant. 19, 20
PSINES	Past SItuations to predict the NExt Situation. 77, 91, 96–99, 106–108
RMSE	Root Mean Square Error. 121
RNN	Recurrent Neural Network. 39
SCRAFFT	Sensors CRAFFT. 91, 92, 94, 96–98, 101, 102, 105–107
SNN	Siamese Neural Network. 38
SVM	Support Vector Machine. 38, 52–55, 58, 62, 64–66, 69–73, 86, 90, 99, 100
SWRL	Semantic Web Rule Language. 36

Mathematical Notations

Sets

$\mathbb{R}$	The set of real numbers.
$\varnothing$	The empty set.
$[\![a, b]\!]$	The set of integers from a to b, $a \leqslant b$.
2^x	The power set of the set x.
$\mathcal{I}$	The set of identities in the home.
$\mathcal{T}$	The set of timesteps in the home.
$\mathcal{P}$	The set of places in the home.
$\mathcal{A}$	The set of activity classes in the home.
$\mathcal{A}^{(i)}$	The set of activity classes in the i^{th} place.
$\mathcal{S}$	The set of sensors in the home.
$\mathcal{S}^{(i)}$	The set of sensors in the i^{th} place.
$\mathcal{C}$	The set of correspondents.
$\mathcal{M}$	The set of communication modalities.
$\mathcal{A}v$	The set of possible availability values.
$\Delta^{(i)}$	The set of decisions taken by the classifier of the i^{th} place.
$\overline{\Delta}$	The set of fused decisions.

VALUES

$\delta_{k,j}^{(i)}$	Decision of the classifier of the i^{th} place about the j^{th} activity class of the k^{th} place.
$\overline{\delta}_{j}^{(k)}$	Fused decision about the j^{th} activity class of the k^{th} place.

OPERATORS

$\|x\|$	The cardinality of the set x.
$x^{\intercal}$	The transpose of matrix x.

List of figures

List of tables

CHAPTER 1

INTRODUCTION

AUTOMATION has been an important vector of progress in work efficiency, scientific research, and quality of life. Clocks or mechanical calculators, which are some of the first examples of automated systems, allowed people to measure quantities or to perform calculations much more efficiently and accurately. Advances in information technologies and computer science have allowed the development of automation strategies on more complex subjects. In particular, the increasing sensing and computing capabilities of everyday objects has enabled new perspectives for automation.

One such perspective is that of *home automation* (sometimes called Domotics), where we aim to automate daily tasks, chores, or energy management in the home. Automatic regulation of heating and air conditioning, lighting control, or energy consumption management are examples of use cases first considered and developed when home automation arose. However, these services often did not react properly with respect to users' expectations, and thus became more of a nuisance than a convenience [34]. Such inappropriate automation is a great barrier to adoption for home automation technologies.

One of the commonly given reasons for these inappropriate results is the difficulty of adapting the behaviour of an automation service to the specificities and preferences of a particular household. In particular, it is often difficult to infer the needs of occupants of a home based directly on low-level sensor data. Moreover, automation for the sake of time saving is actually not often the main reason for the adoption of similar technologies in homes: users are commonly more interested in services that improve their quality of life instead [34].

Moreover, automation services are not the only kinds of services that can

be provided in a home. Using the definitions of Crowley and Coutaz in [34], classical automation would often fall into tool service and housekeeping service categories, but not in advisor and media services, which provide information and suggestions to occupants, or extend their perceptual capabilities. As such, the term *smart home* is now preferred to home automation. Under this new denomination, we take into consideration general computer-assisted home services, including home automation itself. In this thesis, we study some of the algorithmic processes required to provide general services in smart homes. We argue that these services require knowledge on the *context* of the home, in order to avoid inappropriateness. In particular, we study algorithmic solutions to obtain information about the activity of occupants in homes (both present and future), which constitutes a major part of context, from typical sensing devices that can be installed in a home.

In Section 1.1, we motivate our thesis work with an example of a communication assistant for the smart home. We show that, in order to provide such a service, it is essential that the home can discover its internal context and know about the activities and availabilities of its occupants. In Section 1.2, we present our contributions on the problems of activity recognition, activity prediction, and availability inference, in relation to our goal of providing a service of communication assistance in the home. Finally, in Section 1.3, we give a short overview of each following chapter of the thesis.

1.1 Communication assistance in homes

One major aspect of the daily life of people is communication. With recent technological advances, the number of communication modalities (through the internet, smartphones, etc.) and thus potential contacts has significantly increased, to the point where incoming communication attempts can become a nuisance. A home that manages incoming communications for its occupants can thus be a valuable service that improves quality of life.

Such a communication assistant could for example suggest appropriate moments for an outsider to call an occupant, based on that occupant's availability to communicate. It could suggest appropriate modalities of communication or devices to reach an occupant: for example, if an occupant and their landline phone are on different floors, the assistant can suggest to call on their mobile phone rather than the landline phone. It could automatically delay the delivery of messages such as e-mails until the occupant is available, in order to reduce their mental load.

In order to implement such a communication assistant, it requires access to information about the availability of occupants for communication. This availability can greatly vary depending on the occupant's identity, their daily routine, the current time, the place they are in, their mood, or even outside events. As such, a communication assistant cannot provide valuable services unless it has access to personal preferences and situations of its occupants; if it did not, it

would probably exhibit inappropriate behaviours and suggestions to outsiders in a manner similar to past home automation systems. Moreover, such a system may need to anticipate future situations of occupants, in order to give suggestions for future availabilities of occupants.

Therefore, a communication assistant is a complex example of smart home service that raises a number of algorithmic problems on context recognition and prediction. We will thus use this example of communication assistance as the motivating service for all our contributions in this thesis. In particular, this service requires activity recognition, activity prediction, and availability estimation capabilities.

Activity recognition The problem of activity recognition in smart homes consists in automatically identifying the current activity of an occupant using only data collected by sensors installed in the home. Smart home sensors typically record low-level data such as temperatures, electrical consumptions, motion detections, door opening events, etc. Activities are complex sets of tasks performed by an occupant with a set goal, such as cooking, showering, sleeping, etc.

Activity recognition must be as accurate as possible, so as to limit the possibility of providing inappropriate services that rely on activity information. However, recognizing such complex activities from heterogeneous and individually poorly informative sensors is difficult. In particular, the relationship between activities and sensor data is highly dependent on occupants' preferences, routines, and moods, as well as the topology of the home and the existing sensor installation.

Activity prediction The problem of activity prediction in smart homes consists in automatically predicting future activities of an occupant using their past situations as well as their current situation. Specific instances of this problem include predicting the next future activity, predicting a sequence of future activities, or predicting the next time of occurrence of each activity class.

Activity prediction, much like activity recognition, must be as accurate as possible, both regarding activity labels as well as time of occurrence of these activity instances. Activity prediction relies of previous situations and thus on an activity recognition step, which is thus an additional source of confusion. Routines of occupants can highly vary from one person to the next, from one day to another, and unexpected changes can occur depending on the mood of occupants, which is generally unobservable.

Availability estimation The problem of availability estimation in smart homes consists in deciding whether a communication attempt from the outside would inappropriately interrupt an occupant or not, based on their situation. We can also anticipate if potential communication attempts at a future time would be appropriate. Availability of an occupant can greatly vary from one situation to the next, and is often difficult to precisely evaluate even for the occupant themselves.

Few works study availability estimation in smart home settings. Availability estimation has indeed been mostly studied in the context of professional environments and directly on smart phones, where availability is more easily decidable and where communication assistance was seen as being valuable for a long time. In smart homes, availability estimation should rely on the accurate identification of situations, including activity. Linking situations to availability in homes is not an extensively-studied problem.

1.2 Contributions of this study

We propose 4 main contributions in this thesis:

Orange4Home We propose a new dataset of labelled activities of daily living in a realistic home setting, which we openly share. Availability for communication of the occupant is also labelled in the dataset. We constructed this dataset such that each algorithmic piece required for our motivating communication assistance service can be evaluated on it. As far as we know, Orange4Home is the only available dataset of availability for communication in smart homes.

Place-based activity recognition We propose a new activity recognition approach that relies on the relationship between context dimensions of place and activity. More precisely, we suggest that instantiating different activity recognition models depending on the place (with their own independent sets of sensors and possible activity classes), independently from each other, will lead to simpler models and thus higher recognition accuracies compared to global approaches. We also argue that such a non-monolithic approach allows modular training phases which is valuable in smart homes and which will shorten computing times. We evaluate the behaviour of our approach on a state-of-the-art dataset and Orange4Home.

PSINES for activity prediction We propose a new activity prediction model called PSINES, which extends state-of-the-art work. We propose to use both context information and sensor information to predict future activities. We propose to model non-Markovian relationships between activity sequences, contrary to most state-of-the-art solutions. We suggest to model the cognitive state of the occupant using a latent unobserved variable. We argue that these 3 extensions should greatly improve activity prediction accuracy. We evaluate the behaviour of our approach on state-of-the-art datasets and Orange4Home.

Availability inference from context We propose to model availability for communication in homes as a function of other context dimensions. We introduce two context dimensions essential to infer availability: the correspondent that initiate the communication, and the modality of communication used. Other important context dimensions for availability estimation in homes include place

and activity. We show that a baseline inference function based on this estimation workflow can achieve high performances on Orange4Home. We evaluate the impact of incorrect activity recognition on availability estimation.

1.3 An overview of the thesis

We present below an overview of each chapter of the thesis.

— Chapter 2 presents our definition of context in smart homes and the data sources that can be used to observe this context. We first survey state-of-the-art definitions of context in various fields of study, from which we give a definition of context specifically for smart home environments. We define in particular what we mean by activity of occupants in homes, which is a major part of the context.

We then discuss the different categories of data sources that are typically used in smart home systems: audio-visual sensors, wearable sensors, and ambient sensors. We discuss the advantages and drawbacks of each type of data source, and argue that ambient sensors should be primarily used. This choice conditions our contributions, which have to adapt to these data types.

Finally, we survey state-of-the-art datasets of activities recorded in smart home environments. We use some of these datasets in our experimental studies. We propose a new dataset of activities of daily living, called Orange4Home, following a discussion on the issues commonly found in state-of-the-art datasets.

— Chapter 3 presents our contributions to the problem of activity recognition. We first introduce the problem in more details as well as the underlying assumptions we make. We then survey previous works in the literature on this problem and discuss some of the drawbacks that appear among them.

Following these first 2 sections, we present our place-based activity recognition approach, and our motivations for its design. We discuss the main advantages of this approach compared to state-of-the-art approaches. We emit a number of hypotheses that we expect to verify experimentally.

We finally present a number of experimental results on the performances and computing times of the place-based approach compared to global approaches. We use the Opportunity dataset found in the literature [128] as well as Orange4Home, for these experiments. We confront our hypotheses to these results.

— Chapter 4 presents our contributions to the problem of activity prediction. The word “prediction” has many meanings in the literature. We provide a proper definition of prediction, and present the assumptions we make about the problem. We survey the state of the art of activity prediction (in

the sense that we previously defined) and identify in particular a promising model that we think can be substantially improved.

From this model, we propose 3 main extensions to improve prediction performance. We discuss the motivations for each of the 3 extensions, and propose a combined model called PSINES. We emit a number of hypotheses that we expect to verify experimentally on the behaviour of PSINES.

We finally present a number of experimental results on the prediction performance of PSINES, intermediate models, and state-of-the-art approaches. We use a group of 5 related datasets from the CASAS project [32], as well as Orange4Home, for these experiments. We confront our hypotheses to these results.

— Chapter 5 presents our contributions to the problem of availability estimation. We define availability estimation as the problem of evaluating whether occupants of homes are willing to be interrupted by remote communications. We then survey state-of-the-art works on availability inference in professional environments, on smart phones, and in smart homes.

We then propose to model availability as a function of other context dimensions, which include in particular place, activity, correspondent, and modality of communication. We present a workflow for inferring availability following activity recognition or prediction (using our previous contributions on both of these problems), and propose a simple averaging inference approach as a baseline. We discuss on what domains of values should be assigned to the context dimensions of correspondents, modalities, and availabilities.

We finally present experimental results on availability inference following an activity recognition step (using place-based activity recognition). We use the Orange4Home dataset for these experiments, as it is the only smart home dataset containing labelled availabilities, as far as we know. We study the impact of the activity recognition step on availability inference.

— Chapter 6 concludes the thesis with a summary of our contributions, and their impact on future work. We finally discuss the limitations of our contributions and propose some potential perspectives to address these shortcomings.

CHAPTER 2

Context in the home

Providing context-aware services to occupants of a smart home presupposes that this smart home is able to discover and maintain knowledge of said context. In order to properly implement general public smart home systems with such context-aware capabilities, it is first and foremost necessary to precisely define what we mean by context. This is the goal of Section 2.1, in which we clarify in particular the role of occupants' activities as context information. Following this definition, we survey in Section 2.2 the main categories of sensors that can be used to collect data about the home, and from which algorithms can infer the activity of occupants. We discuss which of these categories is well-adapted to general public smart home systems. Finally, we present in Section 2.3 the first contribution of this thesis: Orange4Home, a dataset of daily living activities in the home, following an analysis of the characteristics of similar state-of-the-art datasets.

2.1 Modeling context in the home

In this section, we propose an explicit definition of what context is in homes, as well as related terms. In particular, we explain how the activity of occupants is one of the keystones of context information according to our definition, and thus that it is fundamental for a smart home system to be able to automatically recognize these activities so as to be context-aware. This definition work, presented in Section 2.1.2, is preceded by a survey of state-of-the-art definitions of context in Section 2.1.1, which constitute the foundation of our proposed definition.

2.1.1 Context as defined in the literature

We present in the following subsections 4 significantly different definitions of context from various fields of computer science, as well as examples of how those definitions apply to a situation of living in a home. An extensive survey of state-of-the-art definitions of context is proposed by Bazire and Brézillon in [15].

2.1.1.1 Context as the location and changes of nearby people and objects

In [132], Schilit and Theimer first introduced the notion of *context-aware computing*. In this paper, they report that the spread of new mobile objects with computing capabilities, as well as the emergence of distributed computing on those mobile objects, lead to a new paradigm of computer interactions and executions. Indeed, contrary to the relatively fixed environment of a personal computer, the environment of execution and interaction with mobile devices is greatly dependent on the types and locations of such devices.

Consequently, the definition of context given by Schilit and Theimer (and as reformulated by Crowley et al. in [35]) is the following:

Definition 2.1 (context according to [132], as reformulated by [35])**.** *"[Context is] the location and identities of nearby people and objects and changes to those objects."*

Unsurprisingly, this definition is grounded in the field of distributed mobile computing, as evidenced by the restriction of context to *nearby* (in the physical sense) people and objects.

> **Example: Context of a home occupant using Definition 2.1**
> *Jane Doe is watching TV on the couch in her living room. Context in this example is thus that Jane Doe, her TV, her TV remote, and her smartphone are all located close to each other in the living room.*

2.1.1.2 Context as computationally available information

Hirschfeld et al. present in [66] a new programming approach, context-oriented programming, in response to the lack of clear programming designs to use in current programming languages so as to address the need for programs to adapt their behaviour to their execution context.

Context in context-oriented programming is defined as such:

Definition 2.2 (Context according to [66])**.** *"Any information which is computationally accessible may form part of the context upon which behavioural variations depend."*

Therefore, according to this definition, context is limited only to information which may be captured and then transmitted to the program. This implies that

context is closely tied to the sensors that are used to capture it; information that cannot be captured by the system (due to a lack of instrumentation), and information that no sensor is known to capture (due to a lack of scientific and technological advances) are thus not considered to be part of context.

Example: Context of a home occupant using Definition 2.2
Jane Doe's TV can send various information to the smart home system: that the TV is on, that the current channel displayed is Channel 1, and that the sound volume is at 42%. Those 3 pieces of information constitute Jane Doe's context, assuming no other element sends information to the smart home system.

2.1.1.3 Context as a subjective view of the world

Giunchiglia argues in [57] that context is both a local and a partial view of the state of an individual. He proposes the following definition of context:

Definition 2.3 (context according to [57])**.** *"Context is a theory of the world which encodes an individual's subjective perspective about it."*

This definition relies on the viewpoint of Giunchiglia that the context of an individual is used by this individual to *reason* about a goal. Therefore, this context is necessarily limited to the sensing capabilities, the modelling capacities and the reasoning abilities of that individual. It is thus subjective and incomplete. The complete, objective state of the world at a specific moment in time is called a *situation* by Giunchiglia.

Following this definition, Giunchiglia et al. model context in practice as the union of 4 different pieces of information [58]:

- the temporal context (i.e. your current activity);
- the spatial context (i.e. your location);
- the social context (i.e. the other people you are with);
- the object context (i.e. the objects you are with).

Example: Context of a home occupant using Definition 2.3
Jane Doe's personal context comprises the following 4 aspects: her temporal context is that she is watching Channel 1 on her TV; her spatial context is that she is on the couch in her living room; her social context is that she is alone; her object context is that she is near her TV, the remote of her TV, and her smartphone.

2.1.1.4 Context as characterization of situations of entities

In [46], Dey and Abowd survey previous definitions of context and context-awareness used in the literature of ubiquitous computing. They notice that, although most researchers have an approximate idea of what context is, context

is generally not a well-defined term. In particular, Dey and Abowd consider the definitions of context given in the few articles they surveyed to be too specific: they are either definitions based on examples, which makes them difficult to apply to new applications, definitions based on synonyms, which shift the burden of proper definition on those synonyms, or definitions based on specific applications, which makes them non-generalisable (Definition 2.1 and Definition 2.2 are examples of such specific definitions).

Following those observations, Dey proposes in [45] this very general definition of context:

Definition 2.4 (context according to [45])**.** *"Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."*

Here, as opposed to Definition 2.3, there is no notion of locality or subjectiveness. Any information, whether it is known by the entity or not, that can be used to characterize its situation in the world, is context information. In addition, Dey makes it clear with his definition that context information can be, to him, attached to any entity in general; Definition 2.3 uses the term "individual" which suggests context can only be attached to persons.

Based on this definition, Dey and Abowd argue in [46] that there are 2 main levels of context information: *primary* context constitutes the first level, and everything else the second level. Primary context comprises the following elements: *identity*, *time*, *location* and *activity*. These levels are introduced because, according to Dey and Abowd, context information in the second level can be obtained from one or more pieces of primary context information.

> **Example: Context of a home occupant using Definition 2.4**
> *Jane Doe has 4 primary context attributes while watching TV: her identity is Jane Doe, her location is the living room (and more precisely, the couch in the living room), her time is the date and time indicated by her smartphone, and her activity is that she is watching TV. The channel she is watching, which is another piece of context information, can be indexed on her activity of watching TV.*

2.1.2 Our definition of context

The definitions of context presented in Section 2.1.1 have varying degrees of applicability in smart home systems. Definition 2.1 and Definition 2.2 are domain-specific definitions (respectively within mobile computing and within programming languages) that are rooted to examples of those domains. As mentioned in Section 2.1.1.4, such example-based definitions are difficult to apply to other domains and thus to smart home systems.

Definition 2.1 does not consider for example *time* to be part of the context, although it is obvious that the period of the day (the morning, the night, etc.)

can have a big influence on the global context of a home. This definition is thus too specific to be applied to smart homes.

Definition 2.2 includes all computationally available data as a potential part of the context; computational availability is indeed a limitation that a smart home system faces on the context it manages. This definition does not say anything about what the data actually are and from what they are generated. This definition is thus not very informative when applied on smart homes.

Definition 2.3 limits context to what is known by people and is as such not applicable to smart homes. Indeed, the home may need to provide services, such as housekeeping services as described by Crowley and Coutaz in [34], that require knowledge of the context of the home regardless of whether it is currently occupied. Therefore, the context managed by the home is not limited to the subjective view of one of its occupants, contrary to this definition; the home can have a more extensive knowledge of the current context than its occupants.

Definition 2.4 is general enough that it can applied to smart homes. However, in this definition, there is no clear distinction between types of information, the set of those types, or the actual values received by the system. They are all grouped under the word "context", which is limiting when talking about smart home systems. Indeed, types of information that can be used by smart homes, such as the identity of an occupant or their activity, are often determined by the sensors and the algorithms that are deployed in the home. In particular, information about the activity of occupants relies on activity recognition algorithms, which is the main subject of this thesis. Moreover, types of information are often interdependent: for example, your current time depends on your location (because of time zones), or even on your movement if your are travelling at a relativistic speed. Therefore, explicitly defining those types of information independently from the set of those types is important, as we cannot discuss their interdependences and uses in algorithms if they were all defined to simply be "context".

Following those observations, we propose the following Definitions 2.5, 2.6, and 2.7 of *context dimensions*, *context* and *situations* for smart homes:

Definition 2.5. *A context dimension in the home D_i is a type of information that can be used to characterize the situation of a home entity. A home entity is an occupant, a visitor, a pet, a sensor, an actuator, an object of the home, or the home itself.*

Definition 2.6. *The context of a home entity is the set $\{D_1, \ldots, D_n\}$ of context dimensions of this entity.*

Definition 2.7. *A situation $\{d_1, \ldots, d_n\}$ is a specific instance of context, where d_1, $\ldots$, d_n are set values for each of the context dimensions.*

Example: Context of a home occupant using Definitions 2.5, 2.6 and 2.7
Context dimensions that are relevant for us to describe Jane Doe's situation include her identity, the current time, her location in the home, and her activity (other dimensions such as the ambient temperature could be included if a service required

> *such information). The set of those dimensions is the context of Jane Doe that the smart home system will manage. {Jane Doe, evening, living room, watching TV} is the current situation of Jane Doe in the home.*

We will use Definitions 2.5, 2.6, and 2.7 when talking about context dimensions, context and situations in the rest of this thesis. In the following subsections, we discuss in more details on context dimensions in homes: in Section 2.1.2.1, we present 4 primary context dimensions that are important to characterize the context of a home occupant. Activity, which is one of those primary context dimensions, has many different definitions that we aim to unify in Section 2.1.2.2.

2.1.2.1 Primary context of occupants: {identity, time, place, activity}

Although context can comprise, in our definition but also in state-of-the-art ones, any context dimension that is a relevant source of information to provide context-aware services, it often contains in practice a limited set of specific context dimensions. For example, as presented in Section 2.1.1.3, only 4 context dimensions (temporal, spatial, social and object) are used in practice to model context in [58]. Dey and Abowd argue in [46] that context-aware applications typically require the answer to the following 4 questions about the entities they need to serve:

— *who* are you?

— *when* are you?

— *where* are you?

— *what* are you doing?

This leads them to assert that, in practice, 4 context dimensions (which they name *primary context types*) are more important than others: *identity*, *time*, *location*, and *activity*.

Although it is possible to imagine services that do not require knowledge about some of those context dimensions (for example, a basic automatic lighting control system may not necessarily need to know the identity of the person entering the room), it indeed appears that those 4 context dimensions are very often necessary to build context-aware services in smart homes. Understanding routines of living of occupants, which is the main subject of this thesis, is fundamental for many of such services. The routine of an occupant (which we first need to *identify*) is the sequence of *activities* in *time* of that occupant, throughout the various *places* of their home.

Therefore, we will adhere to the assertion of Dey and Abowd, fitted to the subject of this thesis, that identity, time, place (or location), and activity are essential context dimensions to provide context-aware services to occupants of smart homes. We define this *primary context*, using previous Definitions 2.5 and 2.6, in Definition 2.8:

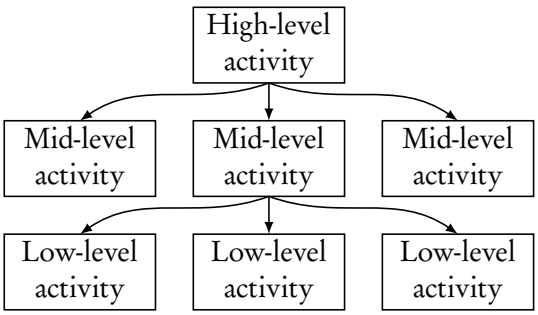

Figure 2.1 – Hierarchical structure of an activity in smart home research.

Definition 2.8. *The primary context of occupants in smart homes is the set of primary context dimensions {identity, time, place (or location), activity}.*

2.1.2.2 What is the activity of an occupant?

The word "*activity*" is very common in regular speech as well as in scientific discourse. As such, it is a very overloaded word that is not always properly defined. In the following section, we report definitions of "*activity*" (and related terms) in smart home research as well as other fields of computed science which deal with activities. We then propose a unified terminology that we will use for the rest of this thesis.

Smart home In parts of smart home research, activities are ranked by levels: authors usually distinguish low-level activities (e.g. walking, sitting) from high-level activities (e.g. cooking, watching TV) [75]. Some authors also introduce mid-level activities [128]. The general idea is that activities of higher levels are constituted of activities of lower levels (see Fig. 2.1).

High-level activities in smart home research are often described as Activities of Daily Living (ADLs). In datasets and various studies, ADLs are chosen based on what was chosen in previous works and don't always involve such level-based hierarchies.

Activity theory According to Kaptelinin and Nardi, activity is defined, in activity theory, as "*a unit of subject-object interaction defined by the subject's motive*" [73]. Activities are thus motivated by the needs the subject has for the object (i.e. their motive).

Activities in activity theory can be represented as hierarchical structures with 3 layers: the activity layer, the action layer, and the operation layer [73]. In activity theory, activities are thus composed of actions, which are themselves composed of operations (see Figure 2.2).

An *activity* is directed by a *motive*, that is the object that the subject wants to attain. A motive may not be immediately conscious; making motives conscious

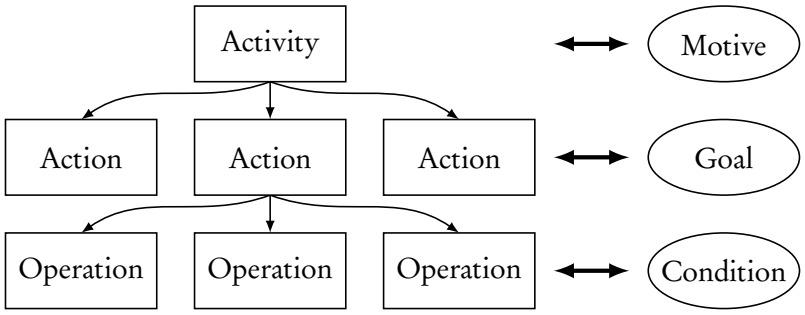

Figure 2.2 – Hierarchical structure of an activity in activity theory.

can require special effort from the subject [91]. An *action* is directed by a *goal*. A goal is conscious: the subject is aware of the goal they want to attain. An *operation* is directed by a *condition*. A condition is relative to the goal of the parent action, that is, the goal of an action must be attained under a set of conditions, which are attained by the operations that constituted the action. The subject is typically unaware of operations. Operations can be improvised.

As the subject learns and repeats certain actions, some of those actions can become operations. This process is called *automatization* (and the opposite process *deautomatization*). The main difference between actions and operations in activity theory is thus that operations are automatized (and thus mostly unconscious) [73]. Similarly, an action can become an activity, and thus a goal can become a motive.

Computer vision In a survey paper by Moeslund et al. [103] on vision-based human motion analysis, it is mentioned that words such as *actions*, *activities*, *simple actions*, etc., are often used as synonyms of each other, with little care from authors in the field. They thus propose to use the following hierarchy: action primitives are atomic movements at the level of limbs; actions are sets of action primitives which describe whole-body movements; activities are sets of actions which give interpretations to what the subject is doing (see Figure 2.3). This hierarchy is also used in [121], another more recent survey paper in the field of computer vision.

Automated classical planning In automated planning theory, the problem of planning is to establish a *set of actions*, corresponding to a sequence of *states transitions*, to reach a *goal state* from an *initial state* [56]. This set of actions is called a *plan* [56] or a *task* [55] (see Figure 2.4). Activity in this field is a frequently used word but does not seem to be a well-defined term with a unique meaning.

Hierarchical task network planning In hierarchical task network planning, the problem of planning is to establish a *set of tasks*, to reach a *goal state* from an

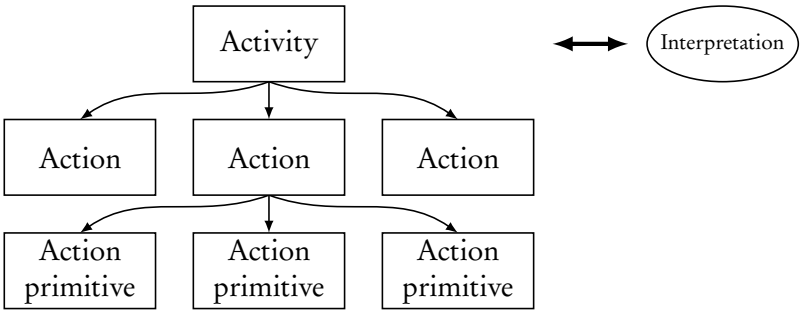

Figure 2.3 – Hierarchical structure of an activity in to computer vision.

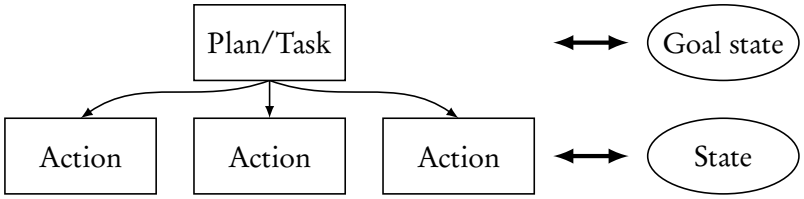

Figure 2.4 – Hierarchical structure of a plan in planning theory.

initial state [56]. Each task is composed of other sub-tasks, except for *primitive* tasks. The root of those tasks is called a *plan* or simply a *task* (see Figure 2.5).

A unifying view As we can see, the definition of an "*activity*" varies heavily from domains to domains, and also from person to person inside the same domain. In order to clearly specify what is meant by activity, tasks, and related terms in this thesis, we thus propose in Table 2.1 a unified hierarchical model of activity, which we will use in the rest of this thesis, put in correspondence with the previously presented hierarchies.

In this unified model, an *activity* is a set of *tasks* directed by an *activity goal*. The subject of the activity is consciously or unconsciously aware of the activity goal. Nonetheless, the activity goal can always be explicited by the subject, with

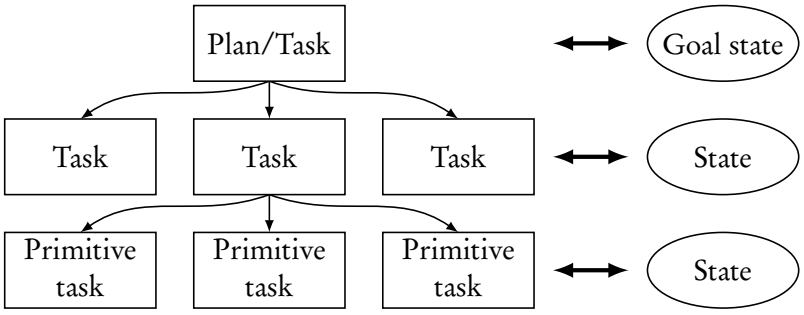

Figure 2.5 – Hierarchical structure of a plan in hierarchical task network planning.

<table>
<tr><th>Unified view</th><th>Smart Home</th><th>Activity Theory</th><th>Computer Vision</th><th>Classical planning</th><th>HTN planning</th></tr>
<tr><td>Activity (activity goal)</td><td>High-level activity/ADL (?)</td><td>Activity (motive)</td><td rowspan="2">Activity (Interpretation)</td><td rowspan="2">Plan/Task (goal state)</td><td rowspan="2">Plan/Task (goal state)</td></tr>
<tr><td>Task (task goal)</td><td>Mid-level activity (?)</td><td>Task (goal)</td></tr>
<tr><td>Action (action goal)</td><td rowspan="2">Low-level activity (?)</td><td rowspan="2">Operation (condition)</td><td>Action (?)</td><td rowspan="2">Action (state)</td><td>Task (state)</td></tr>
<tr><td>Gesture (gesture goal)</td><td>Action primitive (?)</td><td>Primitive task (state)</td></tr>
</table>

Table 2.1 – Comparison of the terminology used for hierarchical models of activity in different fields of research, as well as the unified terminology used in this thesis.

some effort (i.e. the subject can understand and explain why they perform this activity). The activity goal is set, consciously or unconsciously, by the subject.

A *task* is a set of *actions* directed by a *task goal*. The subject is consciously aware of this task goal. The task goal is either set by the following task goals in the activity (i.e. some tasks need to be completed before doing other tasks) or by the activity itself (i.e. some tasks need to be completed to perform an activity).

An *action* is a set of *gestures* directed by an *action goal*. The subject is consciously aware of the action goal. The action goal is set by the environment of the subject.

A *gesture* is an atomic movement of the subject directed by a *gesture goal*. The subject is not aware of the gesture goal but rather performs it automatically. The gesture goal is set by the environment and the morphology of the subject.

Example: Cooking in the home

Cooking *is an activity directed by the goal of* obtaining edible food that I like *(which is a goal I knowingly set myself). To do this activity, I need to perform a set of tasks, among which there is the task of* boiling water. *This task is directed by the goal of* obtaining boiling water, *which is a necessary condition to perform the next task,* cooking pasta *(which is itself directed by the* cooking *activity). To boil water, I need to perform certain actions, such as* filling a saucepan with water. *This action is dependent on my environment, that is the saucepan and water tap of my kitchen. To complete this action, I need to perform certain gestures, such as* grabbing the saucepan. *This gesture is dependent on the saucepan and my own ability to grab it. I am not consciously aware of what I need to do to grab a saucepan.*

2.2 Capturing activities in the home

The ability of a smart home system to recognize and predict the activities of its occupants heavily relies on the sensors it has access to in order to observe those activities. We make the basic assumption that similar sensor data will represent

similar situations, such that smart home sensors can be used to discover context and thus activities in particular [133]. With an inadequate sensors environment, it is possible that certain classes of activities would be undistinguishable from each other using all available data. It is even possible that certain instances of activities would take place in parts of the home not covered with sensors.

Nevertheless, over-instrumenting homes is not a good solution to this problem: sensors can be expensive, not well accepted by occupants, and having too much data can pose algorithmic problems. In the following sections, we present 3 main categories of sensors that are typically used in smart home systems: cameras and microphones, wearable sensors, and ambient sensors. We discuss on which of these categories is the better candidate to collect data in a general public smart home system.

2.2.1 Cameras and microphones

Understanding scenes and identifying activities of people from images and video streams is one of the many applications of computer vision [141]. Recognizing activities of occupants in smart homes is thus a research problem that can be approached from this angle, especially considering how active and large the current computer vision research community is. For example, in [169], ADLs are recognized from RGB-Depth cameras that are placed at different fixed viewpoints and under different lighting conditions. On the other hand, in [120], ADLs are recognized from a camera mounted on the chest of occupants, so that videos are captured with a first-person viewpoint.

In much the same way, recognizing activities from sound is also an approach used in smart home research. We find numerous examples such as [134] or [157] where microphones are used to captured audio streams throughout the home, used as the sole data collection modality to recognize activities of occupants.

In recent years, various cognitive assistants, such as the Google Home or the Amazon Echo, have been introduced in the consumer market. Such systems always include microphones, but also sometimes even cameras [50]. As such, these cognitive assistants can, in addition to their standard uses, be used as data sources for activity recognition.

2.2.2 Wearable sensors

Wearable sensors have been prominently featured in healthcare and elderly care research in homes, where they can be used to conveniently collect physiological data such as heart rate, blood pressure, etc. [126]. Inertial sensors (i.e. accelerometers and gyrometers) are also commonly used: for example, in [67], such sensors help identify body postures, and wearable RFID tags are used to capture hand activities. In general, wearable sensors are often used in conjunction with non-worn sensors, as in [156]. Recognizing activities, postures or location from wearable sensors, as in [88, 119], is also an active research subject.

Inertial sensors are nowadays commonly found in smartphones. This has thus motivated researchers to use them as wearable sensors for activity recognition in general, as in [81]. Since smartphones are often carried by people in their homes, they could thus be included as potential wearable data sources for smart home systems.

2.2.3 Ambient sensors

The last main category of data sources is that of ambient sensors. These sensors are placed at fixed points in the home and typically capture atomic, low-level data about the home. Motion detectors, temperature probes, plugs which measure their energy consumption, door opening sensors, connected light switches, etc., are example of ambient sensors which record physical and state information about appliances, rooms, and the home in general [10].

Although such sensors were historically used in home automation systems, where the simple data they produce could be used to trigger actions automatically in the home, there is growing interest in using such modalities in activity recognition systems. Indeed, these sensors are often small and not as expensive as the previous two categories; as such, covering the entire home with data collection capabilities is possible with ambient sensors. For these reasons, state-of-the-art approaches of activity recognition in the home rely on these data sources, as in [145, 80, 93].

2.2.4 Discussion

As illustrated in the previous sections, each of those 3 categories of sensors are commonly used as data sources for activity recognition in homes. Some studies even make use of more than one of these categories of sensors: for example, in [117], both a wearable headset containing a 3-axis accelerometer, and a video camera, are combined to enhance activity recognition performances. However, it is not clear from these examples that all 3 categories of sensors are equally acceptable data sources for our study, which focuses on general public smart home systems.

Indeed, many examples we previously mentioned are aimed towards Ambient Assisted Living (AAL) for the elderly and for people with medical conditions. Such applications are significantly different from general public smart home services. Consequently, the suitability of the 3 categories of sensors can also significantly vary for our purposes.

2.2.4.1 Acceptability

Acceptability is one of the main problems of home instrumentation: privacy concerns can indeed lead people to reject smart home technologies. Since sensors are the most physically visible parts of a smart home system, they can often be the first reason that motivates people to reject such technologies. Townsend

et al. show in [146] that, in the context of AAL, wearable sensors measuring physiological data are typically better accepted than wearable sensors, while cameras were the least accepted category of sensors. Debes et al. draw the same conclusions in [42]. Indeed, they argue that the first two categories of sensors capture significantly less sensitive data compared to cameras, which makes the trade-off between privacy and usefulness fair for occupants. Similarly, complete audio streams can contain sensitive conversations that occupants would not want recorded.

For general public smart home systems, these acceptability concerns are very similar. In fact, it is likely that the threshold for rejecting sensors is even lower than in AAL applications, as occupants do not have health concerns that would make them accept worse trade-offs between privacy and usefulness of collected data. In particular, this means that wearable sensors, especially those collecting health-related data, are likely to be less accepted in general public populations than in typical AAL populations.

2.2.4.2 Expensiveness

Another common concern in the adoption of smart home technologies, especially for the general public, is that of the cost of such installations. Lara et al. argue in [84] that using ambient sensors in smart homes implies significant installation and maintenance costs. Indeed, since these sensors typically capture very little information by themselves, a high number of such sensors is required to comprehensively capture activities in the home. Installation and maintenance is thus time consuming, and the required mass of sensors counterbalances their individual costs. However, we can hypothesize that the cost of such sensors will decrease as smart home systems democratize. Similarly, it is possible that future homes will be directly instrumented during construction, thus reducing installation costs.

On the other hand, smartphones are expensive systems compared to ambient sensors. Moreover, not every occupant of a home owns one (e.g. children). As such, using smartphones as wearable sensors is problematic as they would potentially not cover every occupant in a home.

Cameras and microphones are quite expensive compared to the other two categories of sensors. High definition, in terms of image or of audio, are most likely required to properly capture activities in the home, which increases costs even further.

2.2.4.3 Conclusion

Among the 3 main categories of sensors typically used in smart home systems, it appears that ambient sensors currently are the most likely to integrate well in systems aimed at the general population. They capture atomic, low-level data which considerably lowers privacy concerns (especially compared to cameras

and microphones). They are less cumbersome than wearable sensors, which is important for occupants of the general public who do not have health issues that would make them accept to wear such sensors more willingly. Ambient and wearable sensors are generally less costly than vision and sound based data collection systems.

For these reasons, we will, in this thesis, study the problem of activity recognition and prediction from ambient sensor data, or occasionally from a combination of ambient sensors and wearable sensors.

2.3 Datasets of activities in the home

In order to both train and evaluate activity recognition and prediction algorithms, datasets of labelled activities are a necessity. Recording such datasets is an expensive and difficult process. Indeed, very few instrumented homes currently exist; it is thus often required to equip a home with sensors for the purposes of data collection, which is costly and technically demanding. Secondly, labelling activities, often over long periods of time, is a tedious, but crucial process; a dataset with inaccurate labels loses much of its value. Finally, the diversity of home layouts and households is so large that many such datasets should be recorded, in order to get a sufficiently representative sample of activities in homes.

In Section 2.3.1, we present examples of such home activity datasets that are openly shared to the scientific community. We discuss their strong points, as well as their drawbacks in the context of our study. We then present in Section 2.3.2 a new dataset of activities in the home, constructed to fill in the specific needs of this thesis, and that we openly share to the scientific community.

2.3.1 State-of-the-art datasets

We present in this section a number of state-of-the-art datasets available in the literature.

2.3.1.1 Opportunity

The OPPORTUNITY Activity Recognition Dataset [128] is presented as a benchmark dataset for algorithms related to human activity, such as activity recognition or activity segmentation. This dataset comprises two main scenarios, performed by 4 subjects (independently from each other):

— *ADL run*: this scenario consists of the following sequence of activities: *Start*, *Groom*, *Relax*, *Prepare coffee*, *Drink coffee*, *Prepare sandwich*, *Eat sandwich*, *Cleanup*, *Break*. This scenario is performed 5 times by each subject. An example of data collected during such a scenario is presented on Figure 2.6.

— *Drill Run*: this scenario consists of 20 repetitions of the following specific sequence of actions: *Open the fridge*, *Close the fridge*, *Open the dishwasher*,

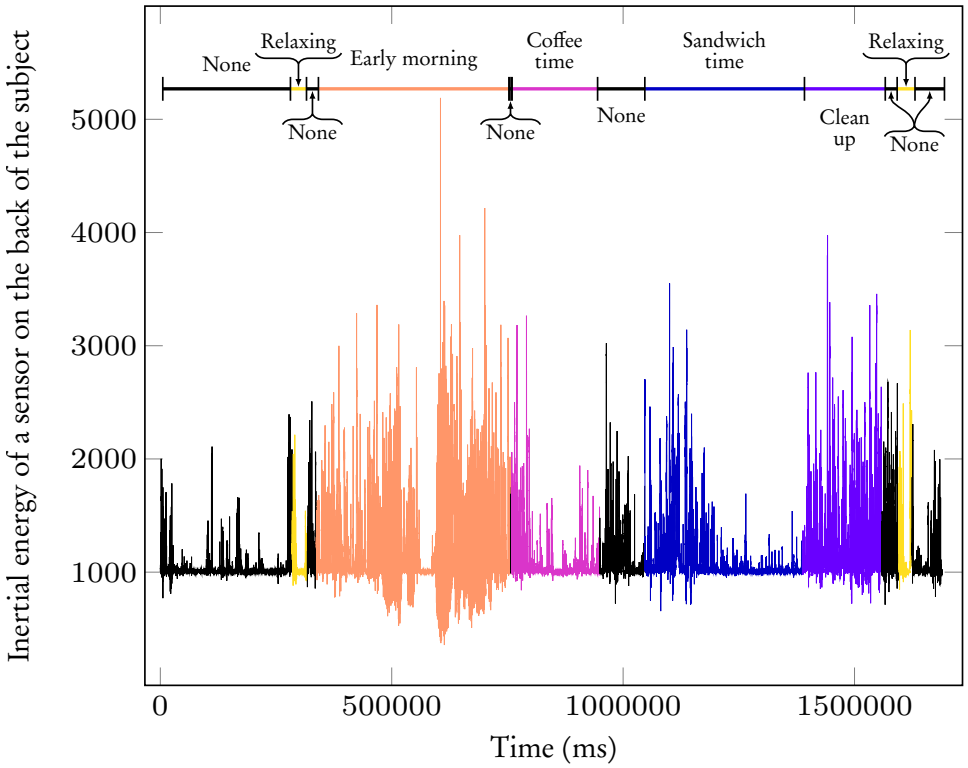

Figure 2.6 – Inertial energy of an accelerometer on the back of a subject during a scenario of data collection in the Opportunity dataset.

> *Close the dishwasher*, *Open drawer 1*, *Close drawer 1*, *Open drawer 2*, *Close drawer 2*, *Open drawer 3*, *Close drawer 3*, *Open door 1*, *Close door 1*, *Open door 2*, *Close door 2*, *Toggle switch*, *Toggle switch*, *Clean the table*, *Drink while standing*, *drink while seated*.

Both of these scenarios took place in a single experimental room (see Figure 2.7) instrumented for the purposes of this data collection, which lessens its representativeness of real inhabited homes. Each subject was equipped with 7 inertial measurement units (3D accelerometer, 3D gyrometer, 3D magnetometer), 12 3D accelerometers, and 4 3D coordinates measurements from a localization system. 12 objects of the room were equipped with 3D accelerometers and 2D gyrometers. Finally, appliances and doors of the room were instrumented: 13 switches and 8 3D accelerometers were used for this purpose. In short, nearly half of all sensors used in this dataset are body-worn sensors, which is not acceptable for systems aimed at the general population, as explained in Section 2.2.2.

Data were synchronized so that each row of input data corresponds to exactly one timestamp; we thus have at each timestep the current value of each sensor. The dataset contains a sizeable amount of missing values, which as explained by the authors are partly due to the loss of messages from wireless sensors.

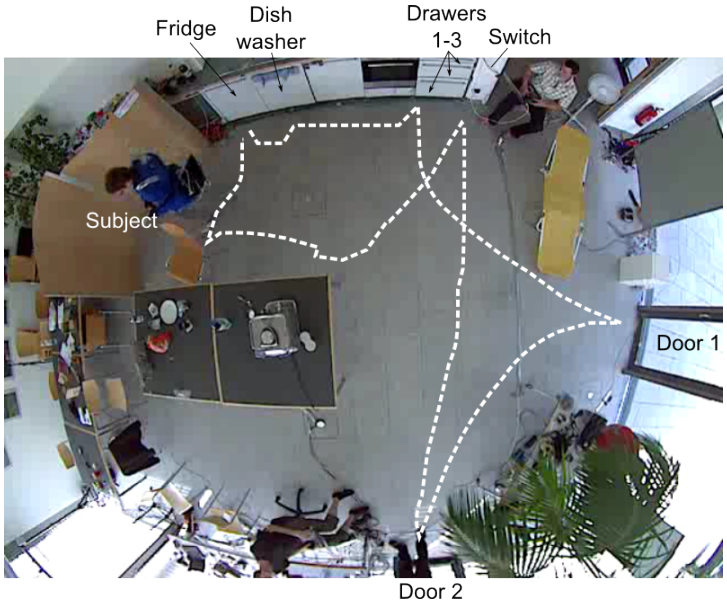

Figure 2.7 – Experimental setting of the Opportunity dataset.

At each timestep, 5 different labels are also provided: the gesture made by the left hand of the subject (e.g. *Reach door*), the gesture made by the right hand of the subject, the action of the subject (which is generated automatically from gestures, e.g. *Open door*), the activity of the subject (of which there are 5 classes, e.g. *Prepare sandwich*), and the modes of locomotion of the subject (e.g. *Sitting*, *Walking*).

2.3.1.2 The CASAS datasets

CASAS is a smart home design and instrumentation kit developed by researchers at Washington State University [32]. The goal of this project is to allow the quick, straightforward, and inexpensive instrumentation of any home with state-of-the-art smart home sensing technologies. As such, more than 30 datasets of daily living captured using the CASAS project are available to the scientific community [1]. These datasets were generally captured in homes of 1 or 2 occupants, and contain the true labels of activities performed by the occupants during the entire duration of the data collection phase.

The instrumentation kit comprises motion sensors, light sensors, door opening sensors, as well as temperature sensors. The datasets recorded using the CASAS kit thus only contain these data sources, which may be limiting when studying the value of various data sources when capturing context in the home. Similarly, this may limit the amount of information captured in the dataset, for example when occupants' activities cannot be distinguished clearly using only these data sources.

2.3.1.3 The Transfer Learning dataset

The Transfer Learning dataset [153] is a dataset that contains the records of 3 subjects (26, 28, and 57 years old) living in 3 different homes (a 3 rooms apartment, a 2 rooms apartment, and a 2 story house respectively). Each home was instrumented with (14, 23, and 21 respectively) sensors for a period of (25, 13, and 18 respectively) days.

Activities were segmented and labelled by each user *in situ*, using a Bluetooth headset for the first 2 apartments, and using handwritten notes for the 2 story house. There are 8 activity classes as well as one extra class for activities that were not labelled by the user:

- *leave house*,
- *take shower*,
- *go to bed*,
- *prepare dinner*,
- *other*.
- *toileting*,
- *brush teeth*,
- *prepare breakfast*,
- *get drink*,

The dataset offers data as a sequence of timestamped sensor events; it has not been synchronized. All sensors provide binary data about an event in the home: reed switches detect the opening and closing of doors, passive infrared sensors detect movement in rooms, etc.

2.3.1.4 Activity recognition in the home dataset

Tapia et al. present in [145] a dataset that contains the records of 2 subjects (2 women, 30 and 80 years old), both living alone in their apartments. The apartment of the 30 years old subject was instrumented with 77 state-change sensors, while the apartment of the 80 years old subject was instrumented with 84 state-change binary sensors (see Figure 2.8). Data were collected for 14 consecutive days in each apartment. This dataset was recorded with the intent of evaluating activity recognition algorithms that can serve healthcare applications, such as elderly care, as opposed to general public smart home systems.

Activities were segmented and labelled by the subjects themselves, using a Personal Digital Assistant (PDA) every 15 minutes while the subjects were at home. 35 different classes of activities are considered in this dataset, based on a set of activity categories proposed by Robinson et al. in [127].

As indicated by the authors, this labelling approach was very coercive to the subjects, who didn't always label their activities when the PDA asked them to every 15 minutes; the subjects also did not at times label the activity properly, or with the correct time period. Therefore, the authors, with the help of the subjects, labelled sizeable amounts of data by hand after the experiment ended, by observing sensors events.

Figure 2.8 – Examples of state-change sensors in one of the 2 homes instrumented for the activity recognition in the home dataset.

2.3.1.5 MavPad 2005

The MavPad 2005 dataset [166] provides the records of 8 weeks of living of a subject in a student apartment. This apartment was instrumented with 76 heterogeneous sensors: lightings usage, temperature sensors, humidity sensors, motion detection, door openings, water leakage sensors, smoke detectors, and CO_2 level sensors.

This dataset does not contain any labelling of activities, tasks, actions, or gestures, and thus cannot be used to evaluate the performances of algorithms related to activity recognition. It is instead used by its authors to evaluate sensor events prediction algorithms, in particular motion detection and interactions with the lighting system. Although those events are heavily correlated to the activities, tasks, actions, and gestures of inhabitants, they do not directly describe those elements and are much less informative than them for context-aware services.

2.3.1.6 Discussion

In general, most state-of-the-art datasets of activities in smart homes are difficult to apply in the context of this thesis for two main reasons: *representativeness* issues and *exploitability* issues. The previously presented datasets highlight those aspects.

Datasets are indeed often not representative of the problems we want to tackle in this thesis, which is general public smart home systems. For example, many research programs on activity recognition were devised for elderly care or healthcare applications: the dataset presented in Section 2.3.1.4 is one such example of dataset. Those applications typically have different requirements than general public applications. In particular, the routines of activities of elderly people or people with health issues can greatly differ from the routines of the general public; incidentally, research in this area is more concerned with detecting deviations from the routine (which could indicate degrading health for example)

rather than recognizing activities that feet the learned routine, which can be used to provide services in systems aimed at the general public. Another example of representativeness issues is that some datasets, for example Opportunity presented in Section 2.3.1.1, are recorded in experimental environments that do not mimic closely the complexity of a real home. As such, the interactions of the user with their environment can vastly differ from interactions in a real home.

Datasets can also be unrepresentative for more technical reasons: for example, some datasets, such as the Transfer Learning dataset presented in Section 2.3.1.3, only provide the recordings of specific types of data (in this case, only binary data is provided). Considering the heterogeneity of types of data and sensors that exist in current smart home technologies, such datasets are not representative of many smart home installations in the general public which can include any of such sensors. Another example of technical representativeness issue is the common presence of body-worn sensors in datasets of activities, as in Opportunity presented in Section 2.3.1.1. Such sensors cannot, in general, be suitable for general public systems, as explained in Section 2.2.2.

Exploitability issues are mostly due to the inherent difficulties of accurately labelling datasets with the activities of the subjects. For example, as reported in Section 2.3.1.4, *in situ* labelling can be very coercive for the subjects; the labelling quality can thus be greatly impacted, which makes the dataset hard to use for algorithm evaluation, as bad results could be in part attributed to inaccurate labels.

The time span of data collection can also be the cause of exploitability issues in this thesis. Indeed, to tackle the problem of activity prediction, it is necessary that the dataset used spans enough time such that the routines of the subject are sufficiently well captured, and such that the algorithm can be tested on subsequent periods of living of the occupant. A dataset such as Opportunity, where subjects were recorded during sessions of approximately 30 minutes, can only be used as a benchmark for short-term action prediction, which is limiting for the purposes of this thesis.

In conclusion, few to no dataset presented in the literature fulfils the requirements imposed by the problems we are looking to tackle in this thesis. Although those datasets were soundly constructed, and possess undeniable qualities, they typically cannot be used to their full potential for our problematic of situation recognition and prediction, in the context of general public smart home systems.

2.3.2 Orange4Home: a dataset of activities of daily living

As shown in Section 2.3.1, it it thus necessary, in this thesis work, to construct a new dataset of activities of daily living in homes that is more well-adapted to the constraints and specificities of our study. To avoid representativeness and exploitability issues, presented in Section 2.3.1.6, we established a list of 5 goals that must be met when recording this new dataset:

G1: record data in a real, liveable home which is as seamlessly instrumented as

possible;

G2: establish realistic routines of daily living of an occupant from the general public;

G3: record data on a relatively long time scale;

G4: equip as many appliances and objects of the home as possible, with as many heterogeneous smart home sensors as possible;

G5: accurately label all 4 primary context dimension for the entirety of the experiment.

The following subsections present the various methodologies adopted for data collection to attain each of these 5 goals: the environment used in Orange4Home to fulfil goal G1 is presented in Section 2.3.2.1; the scenario of daily living and the planning of activities for the 4 weeks of data collection, established to fulfil G2 and G3, are discussed in Section 2.3.2.2; the data sources recorded to fulfil G4 are presented in Section 2.3.2.3; the labelling procedure used to fulfil G5 is explained in Section 2.3.2.4.

This data collection phase resulted in the consolidation of a new dataset of activities of daily living in an instrumented home, called *Orange4Home*. This dataset is freely available on request [5].

2.3.2.1 Apartment

The Orange4Home dataset was recorded within the *Amiqual4Home* project [4]. Amiqual4Home is an equipment of excellence funded by the French Research Agency, which serves as an experimental platform comprising prototyping workshops, living labs, and mobile tools for ambient intelligence research projects. One of those living labs is an instrumented apartment, whose purpose is to provide a realistic home setting in which to perform experiments on smart home systems as well as usability tests in a home environment. In particular, this apartment can be used as a smart home dataset recording setting.

Indeed, the Amiqual4Home apartment is an $87\,\mathrm{m}^2$ two-story home, that has been seamlessly instrumented with sensors. We present on Figure 2.9 and Figure 2.10 the layout of both floors of the apartment, annotated with the name of each place. This apartment was instrumented during renovation works. As such, sensors are well integrated to appliances and furniture, instead of being added on in a way that could change the way occupants interact with the home (because the sensor is physically a hindrance, or because its visible presence changes the perception of the occupant).

This apartment has been used to establish other datasets related to smart home research. In particular, ContextAct@A4H [82] is a dataset of daily living activities recorded for AAL research purposes.

2.3.2.2 Scenario of daily living

Orange4Home only contains the records of a single subject. Since this occu-

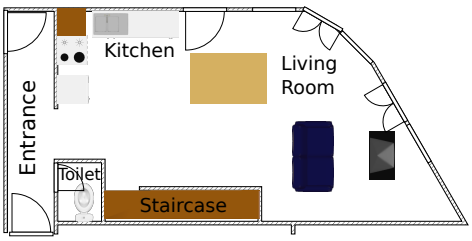

Figure 2.9 – Ground floor of the Amiqual4Home instrumented apartment.

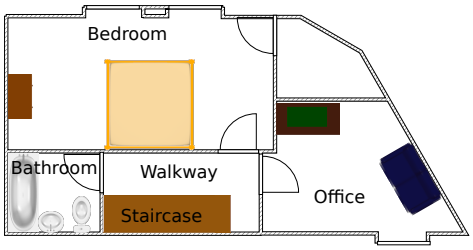

Figure 2.10 – First floor of the Amiqual4Home instrumented apartment.

pant does not live in the Amiqual4Home apartment outside the Orange4Home data collection experiment, and since they will only spend 20 days in the apartment (excluding an acclimatisation day), 3 major issues could occur in the data collection phase:

— the occupant might not use some appliances or some rooms of the apartment that they are not used to or comfortable with, and instead interact with only parts of the apartment that are similar to their own home;

— some activities that are commonly performed in homes might not appear in the dataset because this occupant in particular does not do them;

— some activity classes might have way too few instances at the end of the collection phase, which is a problem in particular if machine learning approaches are used on the dataset.

For these reasons, we deemed it necessary to establish a scenario of daily living in this apartment, as well as an explicit routine of activities for the occupant to follow during the 4 weeks of data collection.

Scenario of daily living in Orange4Home

The apartment is a modern environment in which Bob, the occupant, comes to work alone every working day, from around 08:00 to 17:00. This apartment provides many appliances that allow Bob to not only work, but also have lunch, shower (since Bob comes to the apartment with his bicycle), spend some leisure time, and relax during his pauses. Bob is interested in having personalised services in this environment. As such, he will label his activities for a duration of 20 working days (i.e. 4 consecutive weeks). Bob does not spend time in the apartment outside his working hours (i.e. nights and weekends).

A standard day routine was then established (see Figure 2.11) to fit this scenario: the occupant enters the home; they take a shower and brush their teeth; they go back down to the living room to watch morning news on TV; they then spend the rest of the morning working on their computer in the office; they cook, eat, and do the dishes around noon and then spend some leisure time on their

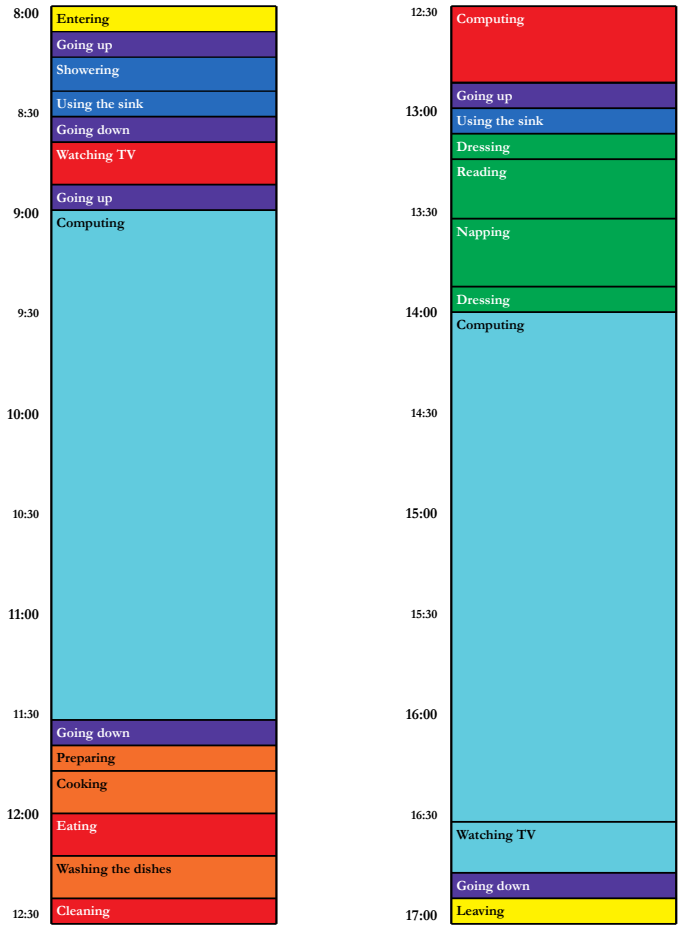

Figure 2.11 – Standard day routine in Orange4Home.

computer; they read and take a nap in the bedroom; they spend the afternoon working again; they briefly watch the news before leaving the home for the day.

In order to increase the variability in the routines of the occupant (and thus the realism and "difficulty" of the dataset), this standard day routine was the routine to follow only for the first two weeks of the experiment; for the last two weeks, the routine of each day was established by applying minor to major changes to the standard day routine (such as interversion, omission, or shortening of activities). In practice, the start and end times of activities in this planning are only indicative. If need be, the occupant could always deviate from the routine and label the actual performed activities (though this did not happen). No instructions were given as to how the occupant should perform any of the activity classes.

The complete list of possible activity classes, grouped by place, in the Orange4Home dataset is the following:

— Entrance: "Entering", "Leaving", "Cleaning";

Place	Data type				Total
	Binary	Integer	Real number	Categorical	
Entrance	3	1	2	3	9
Kitchen	13	21	18	0	52
Living room	16	6	8	7	37
Toilet	3	1	1	0	5
Staircase	3	0	0	0	3
Walkway	9	0	1	0	10
Bathroom	9	6	8	3	26
Office	9	3	3	5	20
Bedroom	17	4	6	7	34
Global	1	13	20	6	40
Total	83	55	67	31	236

Table 2.2 – Number of sensors per place and per type of data in Orange4Home.

— Kitchen: “Preparing”, “Cooking”, “Washing the dishes”, “Cleaning”;
— Living room: “Eating”, “Watching TV”, “Computing”, “Cleaning”;
— Toilet: “Using the toilet”, “Cleaning”;
— Staircase: “Going up”, “Going down”, “Cleaning”;
— Bathroom: “Using the sink”, “Using the toilet”, “Showering”, “Cleaning”;
— Office: “Computing”, “Watching TV”, “Cleaning”;
— Bedroom: “Dressing”, “Reading”, “Napping”, “Cleaning”.

We can note that some activity classes can occur in different places: “Watching TV” can occur in both the Living room and the Office; “Using the toilet” can occur in both the Toilet and the Bathroom; “Computing” can occur in both the Living room and the Office; “Cleaning” can occur in any place. It is indeed not uncommon for occupants to perform some activity classes in multiple different places, which is not a possibility often captured in state-of-the-art datasets.

2.3.2.3 Data sources

As presented in Section 2.3.2.1, the apartment used in Orange4Home is natively instrumented with sensors. Data captured by these integrated sensors include door openings, light switches usage, noise levels, energy consumption of electrical appliances, water consumption, humidities, luminosities, CO_2 levels, motion detections, temperatures, heater settings, etc. Additional information such as weather conditions, outdoor temperature, wind direction, etc., are retrieved from the internet as well. The complete list of 236 sensors can be found at [2].

In addition to the existing pool of sensors, 5 connected plugs were installed for the needs of this particular experiment: one plug on the television in the Living

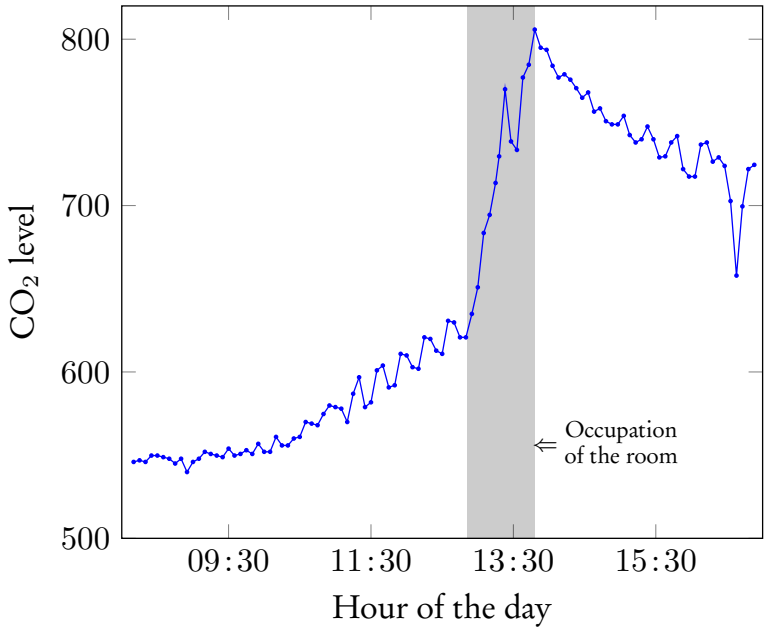

Figure 2.12 – CO_2 levels in the Bedroom on January 31st, 2017.

room, one plug on the television in the Office, as well as 3 available plugs near the table in the Living room, the couch in the Living room, and the desk in the Office, so that the electrical consumption of the laptop of the occupant could be measured during the experiment.

We present in Table 2.2 the number of sensors present in each place of the apartment, grouped into 4 main types of data: binary (e.g. motion, switches), integer (e.g. humidity, water consumption), real number (e.g. luminosity, temperature), and categorical data (weather conditions, heater settings). We can see that the Orange4Home dataset contains truly heterogeneous data sources, as each of these 4 types of data compose a significant proportion of the dataset. There are no body-worn sensors in Orange4Home, for reasons explained previously in Section 2.2.2.

Linking sensor data to activities in Orange4Home is not trivial: sensors by themselves typically provide little information about activities, and that information is often difficult to interpret. For example, we present in Figure 2.12 data collected by the CO_2 level sensor in the Bedroom, on January 31st, 2017. We can see that this sensor reports an abrupt increase in the levels of CO_2 once the occupant is present in the room, which is expected. However, during this time period, the occupant is performing the following sequence of activities: "Dressing", "Reading", "Napping", "Dressing". CO_2 levels by themselves are not informative enough to differentiate between those activities, as the breathing patterns of the occupant are fairly similar for each one of them. Moreover, we see that the CO_2 levels do change even when the person is not present in the room: they increase

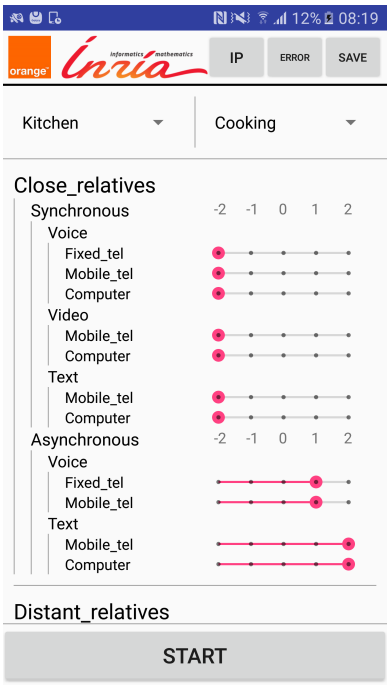

Figure 2.13 – Interface of the labelling application used in Orange4Home.

in the morning, and decrease after the occupant leaves the room. As such, this data can be confusing for an activity recognition system which might mistakenly believe that something is occurring in the Bedroom because of such changes. Finally, we see that the data is quite noisy, which is an additional problem one has to deal with in smart home systems.

2.3.2.4 Labelling activities

As reported in Section 2.3.1.6, it is difficult to label activities both accurately and in a non-burdensome way. Labelling activities *a posteriori* is very costly and error-prone. Indeed, the occupant has a less clear memory of the events happening during each activity instance. Moreover, a large number of ground truth sources (such as cameras) is needed to cover the entire home. On the other hand, *in situ* labelling is tiring and distracting for the occupant, which may impact their behaviour in the home. In particular, the occupant is bound to make labelling errors, especially if the experiment lasts for a long time as in Orange4Home.

As such, we decided to label activities *in situ*, and correct labelling errors *a posteriori*: the occupant carried a smartphone with them, on which was installed the labelling application developed for this experiment (see Figure 2.13). This application allowed the occupant to first select the place they were in, and then the activity they were going to perform in this room (including an activity "Other" for unexpected cases, although this was never used). Pressing "START" when the

Place	Activity	Number of instances	Total
Entrance	Entering	21	42
	Leaving	21	
Kitchen	Preparing	19	61
	Cooking	19	
	Washing the dishes	19	
	Cleaning	4	
Living room	Eating	19	71
	Watching TV	18	
	Computing	15	
	Cleaning	19	
Toilet	Using the toilet	8	8
Staircase	Going up	57	114
	Going down	57	
Bathroom	Using the sink	38	70
	Using the toilet	9	
	Showering	19	
	Cleaning	4	
Office	Computing	46	64
	Watching TV	14	
	Cleaning	4	
Bedroom	Dressing	30	63
	Reading	15	
	Napping	15	
	Cleaning	3	
	Total		493

Table 2.3 – Number of instances of each class of activity in Orange4Home.

activity begun and "STOP" when it ended sent the labelling messages to the data collection system of the apartment.

Using the "Error" functionality, the occupant could write any message, which would subsequently get sent to the system much like activity labels. This functionality allowed the occupant to immediately record any labelling error they had recently made, so that *a posteriori* corrections were much less costly. Video streams captured by 6 cameras (1 in the Kitchen, 1 in the Living room, 1 in the Walkway, 1 in the Office, 2 in the Bedroom) were also recorded during the entirety of the experiment. These video streams provide ground truth on the events happening in the apartment; as such, they were used to accurately correct, in conjunction with the error messages, some of the activity labels of the dataset. The labelling accuracy in the Orange4Home dataset is thus very high.

We present in Table 2.3 a summary of the activity instances labelled in the Orange4Home dataset. We see that some activity classes are more frequent than others (e.g. 46 instances of "Computing" in the Office compared to 9 instances of "Using the toilet" in the "Bathroom"). Similarly, some places are more active than other (e.g. 63 instances of activities in the "Bedroom" compared to 42 instances in the "Entrance"). This lack of balance is representative of real situations, where

some activities are only sporadically performed, and some places are uncommonly visited by occupants.

2.4 Conclusion

This chapter presented the definition of context which we will use in the rest of this thesis. This definition highlights in particular that the activities of occupants constitute a central context dimension. As such, it is necessary for a smart home to be able to automatically recognize such activities to achieve context-awareness.

In order to recognize such activities, the home, much like humans, requires sensors that will allow it to observe such events. We thus surveyed in this chapter the common categories of sensors that can serve as data sources to smart home systems. We showed that ambient sensors are more likely to integrate well in a solution aimed at the general public.

Finally, we presented the first contribution of this thesis: Orange4Home, a dataset of activities of daily living, shared to the scientific community. We constructed this dataset following the identification of characteristics required to properly capture general public smart home data, which state-of-the-art datasets often did not have.

In Chapter 3, we will study the problem of recognizing activities of occupants in homes from sensor data, which is a necessary step to provide context-aware services as shown in this chapter. We will propose a new approach to activity recognition which exploits known information about other primary context dimensions that we previously defined. We will experimentally study the behaviour of this approach on some of the datasets identified in Section 2.3, and in particular on Orange4Home.

CHAPTER 3

Recognizing activities using context

One of the primary context dimensions necessary to provide context-aware services in the home is activity. As such, smart home systems must be able to recognize the activity of their occupants, using data recorded by all sensors installed in the home. In order to provide services that are always well-adapted to the context, this activity recognition algorithm must be as accurate as possible. Based on our preliminary assumptions, presented in Section 3.1, and a survey of the main methods for activity recognition in smart homes published in the state of the art, presented in Section 3.2, we present in Section 3.3 the second contribution of our thesis: the *place-based activity recognition approach*, which seeks to improve the performances and computing efficiency of activity recognition, using the specificities of other context dimensions in the home. We experimentally evaluate this new approach in Section 3.4.

3.1 Problem statement and preliminary assumptions

We seek to design a Human Activity Recognition (HAR) approach that can be applied in smart homes. That is, we need to design an algorithm which, from data collected by sensors typically installed in a home (see Section 2.2), automatically assigns to a sequence of such sensor data an *activity class* (or *label*) that semantically corresponds to the actual activity (as defined in Section 2.1.2.2) performed by occupants and recorded by sensors.

As such, we must confront a number of problems that are inherent to HAR, to smart home environments, or the combination both. In the following section,

we explicitly state some of the hypotheses we make about this problem, in order to limit the scope of the study and make the problem of HAR in smart homes tractable.

3.1.1 Single-occupant situations

Most work published in the literature about HAR in smart homes focuses on single-occupant situations [17], that is, situations where only one person occupies the home at all times (although that person can change). We make the same assumption in this thesis, for the following reasons:

— single-occupant HAR is still not accurate enough to always correctly provide the right service [26, 33, 17]. Further work is thus needed in this area;

— multi-occupant situations seem difficult to analyse regardless of the algorithm used, because the currently available sensors used to record data in homes typically capture raw, general information about the home (such as motion, temperature, electrical consumption, etc.), which will most likely not capture all the subtleties of human interactions that are necessary for multi-occupant HAR.

3.1.2 Information on identity

We make the assumption that the context dimension of identity is given before the activity recognition step. Identification of people is in itself a complete research subject, which is not the focus of our work. We thus assume that the identity of the single occupant in the home is known.

Equivalently, we can assume that the identity of the occupant is not considered important for activity recognition. Although it seems obvious that different occupants will perform activities differently, we can in practice assume that such differences are sufficiently minimal, for most activity classes performed in a home, so as to not impact the activity models used for HAR.

3.1.3 Presegmented activity instances

We assume that the segmentation of an activity instance is given, that is, we know that a sequence of sensor events corresponds precisely to a complete activity instance from beginning to end. Similarly to the single-occupant assumption, most published works also assume presegmented activity instances instead of working on streaming data [80], yet such approaches are still not accurate enough.

We can afford to make this simplifying assumption in the grand scheme of our thesis, because our motivating use-case of communication assistance does not require online HAR (which is the main advantage of not making this assumption). Indeed, future availabilities for communication may occur hours after the current

situation, and the system can thus often afford to wait for the end of an activity instance before recognizing it.

3.1.4 Sequentiality of activities

We assume that no two activities can occur simultaneously, i.e. the occupant never performs two activities in parallel. This assumption is obviously not always verified in practice. However, for most high-level activities, such as the ones defined in the Orange4Home dataset, it is reasonably unlikely that two activities would be performed simultaneously.

3.2 State of the art

Following up on our presentation of the problem of HAR in smart homes, we review in this section relevant literature which exposes state-of-the-art approaches used to tackle this problem. These works will inspire and motivate our activity recognition choices presented in Section 3.3.

HAR is a long-standing problem with multiple domains of applications, such as video surveillance [155], health and sports [12], or Human-Computer Interaction (HCI) [49]. As we have discussed in Chapter 2, Smart homes and AAL also require HAR techniques, since activity is one of the primary context dimensions required by such applications to provide context-aware services.

HAR is a problem whose solution is not only constrained by the domain in which it is applied, but also by the sensors that are available to provide input data. Videos [149, 170], wearable sensors [84], and ambient sensors [28] are typical sources of data used in HAR research. As we discussed in Section 2.2, ambient sensors, and wearable sensors to a lesser extent, are the types of data sources we should use when working on smart home applications.

Most HAR approaches fall in one of two categories: *knowledge-driven* approaches, and *data-driven* approaches. Knowledge-driven techniques typically use logic, or more generally formal reasoning systems, to model the problem at hand; data-driven techniques rely instead on empirical data to model the problem, typically using statistical reasoning [47]. Some HAR approaches make use of both paradigm: we will name these *hybrid approaches*.

In the following section, we present some of the previous knowledge-driven, data-driven, and hybrid contributions in HAR found in the literature. In particular, we survey works which are applied to home environments, which often share the same constraints, presented in Section 3.1, that we will have to take into account. However, we also discuss some HAR approaches that were originally not applied to home environments, but which present valuable insight about the problem of activity recognition in general, and whose contributions can potentially be applied to smart home environments.

3.2.1 Knowledge-driven activity recognition

Constructing Artificial Intelligence (AI) systems using formal reasoning is a fairly old idea: in a 1958 paper, McCarthy discusses the construction of intelligent, learning programs which derive conclusions (and potentially actions) from premises using formal language manipulations [98]. Expert systems and rule-based programming in general are archetypical examples of such knowledge-driven systems, and have thus been extensively studied to solve various AI problems. In such systems, knowledge is represented though a set of facts and rules which can be evaluated on the system's current input data using formal interpretation of the rules (which are typically "if-then-else" expressions). The different conclusions that the system draws from this evaluation enables it to take decisions depending on the situation it faces [60].

In recent years, *ontologies* have been one of the main formalisms used in knowledge-driven systems. They indeed allow to explicitly model the knowledge of such systems with 3 main benefits [158]: firstly, their graphical nature facilitates the implementation of many different kinds of reasoning systems, since it is a well-studied data structure in computer science; secondly, they can be shared between applications through semantic interoperability [61], and therefore allow the construction of more complex knowledge-driven systems which can share knowledge between each other using a common formalism; thirdly, they can be reused and combined, such that, for a particular problem, the ontological knowledge-base of the system can be easily built by picking and merging relevant ontologies into one.

For example, *DogOnt* is an ontology model proposed by Bonino and Corno in [22] which has been designed for home automation environments. Using DogOnt, we can for example model the knowledge we have about a lamp, as shown in Figure 3.1. From this instantiation, a rule-based reasoning system could infer that the lamp is in the first floor of the home (the dashed "*isIn*" edge) from the fact that it is in the living room, which is itself in the first floor (the solid "*isIn*" edges).

As such, knowledge-driven approaches have been applied to the problem of activity recognition, and related smart home challenges [30, 24, 122, 7, 162, 113, 123, 99]. We discuss in more details 3 papers that present such approaches in Section 3.2.1.1, Section 3.2.1.2, and Section 3.2.1.3.

3.2.1.1 Recognizing ADLs using ontological reasoning

Chen et al. present in [29] a knowledge-driven approach for real-time ADL recognition, in which they explicitly model context and activity using ontologies. In particular, they propose to model activities as hierarchical structures where an activity is itself composed of activities. This makes it possible to reason on different levels of activities. Activities can either be abstract (e.g. "Make drink") or specific (e.g. "Make coffee"). Sensor events are properties attached to activity

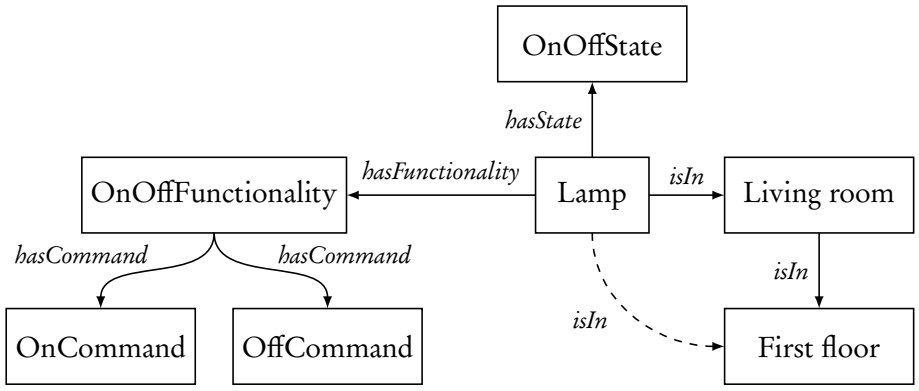

Figure 3.1 – Extract of an instance of a lamp model using the DogOnt ontology.

classes that they can observe.

The proposed activity recognition approach relies on the formalism of description logics [13], in much the same way that the Web Ontology Language (OWL) they use for ontological modelling does. Subsumption is used to recognize activities hierarchically from sensor events. For example, if an instrumented cup signals that it is being filled, and if activity "Make drink" has ontological property "hasContainer", which can be subsumed by the class "Cup", then the system can conclude that generic activity "Make drink" is occurring. Furthermore, if the coffee machine reports that it is being used, the the recognized class "Make drink" can be further subsumed by the specific class "Make coffee".

Extensions of this approach using temporal logic have been proposed by Okeyo and al. in [112].

Chen et al. justify the use of knowledge-driven approaches by stating that there are many common sense relationships between ADLs, sensors, and occupants. As such, explicit ontological models linking these elements should be constructible. However, they state that data-driven approaches typically require large data-sets, which are costly to build and record, and generally do not transfer well from one occupant to another, one home to another, etc.

3.2.1.2 Activity recognition through semantic similarity

Ye et al. propose in [165] a knowledge-based approach for activity segmentation and concurrent activity recognition. This approach employs 3 main ontologies: a domain ontology (which models objects, locations and persons), a sensor ontology (which models sensors themselves as well as sensor events), and an activity ontology (which, in the same vein as the work of Chen et al. presented in Section 3.2.1.1, models a hierarchy of activity classes).

Each activity classes is assigned a set of constraints on time, object, location and persons. More precisely, for a certain activity class, the values of these properties are constrained to certain sets (for example, activity "Cooking" can

only occur in location "Kitchen", and objects manipulated during this activity must be cooking utensils, the stove, etc.). When an input instance is recorded, their approach extracts, from sensor sequences, the semantic features of time, object, location and persons based on their ontological models. The activity class assigned to this instance is then the one for which those semantic features best match its constraints.

Time, object, location, and persons can be seen as context dimensions that this approach uses for activity recognition. Adding constraints to the set of values of context dimensions depending on the current activity is a principle used in some other activity recognition approaches, mostly in hybrid approaches presented in Section 3.2.3.

3.2.1.3 Rule-based presence detection

In [125], Ramparany et al. propose an ontological model of trust in data sources, and uncertainty about information. Each data source is assigned a trust value which quantifies how much we should believe in data it measures, and each information is assigned an uncertainty level which quantifies how much we should believe in that information. Using an Assumption-based Truth Maintenance System (ATMS), they then compute the uncertainties of new information inferred, by applying rules to information measured by the data sources.

Ramparany et al. applied their system to the presence detection problem in instrumented rooms. Rules that model the relationships between sensor events and the presence of occupants are defined using Semantic Web Rule Language (SWRL). Their use-case includes multi-occupant scenarios where the system should recognize the difference between no presence, presence of a single occupant, and presence of more than one occupant. SWRL rules have been defined accordingly, by stating that simultaneous detection of presence in different parts of the room imply that more than one occupant are present. They show that such a system can classify between each of these 3 presence situations fairly accurately using rules, and that it is possible to associate uncertainties to the inferred information of presence using the ATMS.

Further work by Ramparany in [124] gives several examples of information that can be inferred using this approach, such as occupation of rooms or misuse of appliances (e.g. forgetting to close a fridge's door). For such use-cases, a rule-based approach indeed seems easy to implement (as they require few rules that involve few sensors), and applicable to most homes (as they involve commonly found appliances or smart home sensors). On the other hand, this rule-based system seems difficult to apply to the problem of activity recognition, where the design of rules is much more complex than in the previously mentioned examples. Indeed, modelling the relationship between each sensor and each activity class seems unrealistic, especially considering the variability of activity realization from one occupant to the next, and the variability of homes and sensor installations.

3.2.2 Data-driven activity recognition

Data-driven AI finds its roots in the early statistical methods for regression and data fitting, such as the least squares method first published by Legendre in 1805 [90]. Turing, in a classical paper published in 1950 [150], discusses the idea of computer programs that learn to solve problems from experience, rather than from a fixed set of programmed rules. This idea, combined with new developments in statistical methods, lead to the field of study now known as *machine learning*.

In such methods, algorithms are trained on a set of experimentally collected instances that are hopefully representative of the problem at hand. The goal of the training phase is for the machine learning algorithm to build its own model of the training data, based on properties discovered in the training set (such as the correlations between variables), in the hope that these properties hold for any future data instance of that same problem, and can thus be used to process the instance in the same way the algorithm was trained to process them on training data.

The main advantage of such approaches is that model construction is automatically done during the training phase. This is particularly helpful when the underlying problem is too complex to be modelled by experts, in which case knowledge-based approaches become difficult to implement. On the other hand, a major drawback of data-driven approaches, as their name suggests, is that a large amount of data is typically required to properly capture the complexity of the target model that must be learned. Moreover, most of the current well-working techniques are *supervised*, that is, each training instance requires a label that specifies which output is expected for that particular instance (e.g. "Cat" for an image of a cat, if the task consists in recognizing animals in images). Acquiring lots of labelled data is often expensive, sometimes prohibitively so.

Nevertheless, automatic model construction is such a valuable property that data-driven approaches are very popular at the time of writing this thesis. HAR and related smart home challenges, due to modelling complexity, are natural problems to use machine learning approaches on, such as in [8, 114, 14, 116, 33]. We discuss in more details 3 papers that present such approaches in Section 3.2.2.1, Section 3.2.2.2, and Section 3.2.2.3.

3.2.2.1 Activity recognition augmented with past decisions

Krishnan and Cook propose in [80] a data-driven activity recognition approach applied to smart home environments. In this work, they focus on classification of windows of sensor data, rather than fully presegmented activity instances. Recognizing activities from sensor streams rather than complete instances is theoretically beneficial since it means that the system can identify the activity of the occupant before they actually complete their activity, which potentially extends the number of context-aware services the smart home can provide.

However, recognizing activities on fixed-size windows implies a relative regularity in the sampling rates of sensors; it also requires that sensors are sufficiently informative such that an activity can actually be recognized from only a fraction of the entire activity. It does not seem clear that both of these hypotheses are valid in smart home environments.

The construction of feature vectors given as input to the classifier (a Support Vector Machine (SVM) in their experiments) in this work is original: in addition to sensor data, extracted for a specific window, information about the classification performed on the previous window is added. More precisely, a first activity model is trained to recognize activities from sensor data only, and a second activity model is used to recognize the activity of the next window from sensor data as well as classification information on the previous window (e.g. the activity label given to the previous window).

Krishnan and Cook show through this contribution that one can improve the classification performances of an activity recognition approach in the home by augmenting raw sensor data with higher level contextual data, through the introduction of the activity label of the previous activity. This suggests that context information in general can indeed be used in order to improve activity recognition performances. Here, this context information is directly injected into the input of the classifier, in a data-driven fashion. Hybrid approaches, which we discuss in Section 3.2.3, may use this context information differently.

3.2.2.2 Gesture recognition with similarity metrics

In [19], Berlemont et al. tackle the problem of gesture and action recognition using Siamese Neural Networks (SNNs). A classical SNN can be seen as 2 identical Artificial Neural Network (ANN) which are run simultaneously on 2 different instances. If the output vectors of both executions are close (according to some measure), then both instances are member of the class; conversely, if both instances are far apart, the 2 instances are members of different classes. In that sense, a SNN learns a metric of similarity between input instances, unlike most ANN which learn a function between input instances and class labels.

The authors propose to improve the discriminative power of the similarity metric learned by the SNN through modifications of the SNN comparisons. Instead of comparing one instance to a known instance during training or runtime, they argue in favour of comparing examples of instances of all classes simultaneously (as well as the input instance, with unknown class), which should better model the similarity relationships in multi-class problems (such as activity recognition). Berlemont et al. propose a new measure to evaluate the closeness of output vectors, that is more adapted to this new approach.

Through various favourable experimental results, Berlemont et al. showed that multi-class gesture and action recognition problems are better tackled when the classifier used can properly model the relationship between an instance and all possible classes at once, instead of two-by-two sequential comparisons. This

suggests that, for the problem of activity recognition in homes, one must be careful not to block certain relationships between activity classes on the basis of insufficiently correct expert knowledge (such as 2 linked activity classes that occur in different rooms, which we falsely assume don't have any relationship).

3.2.2.3 Action recognition using deep learning

Ordóñez and Roggen present in [115] an action recognition approach based on Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). These ANNs are part of the field of *deep learning* [85], which aims at designing classifiers that not only learn the models from training data, as in classical machine learning, but that also learn to automatically construct the features on which it will learn this model from raw data inputs.

Beyond the contribution on combining CNNs and RNNs, this work demonstrates that deep learning techniques can be applied on wearable sensor data for action recognition, even though the CNN was first designed for image recognition tasks. In particular, this shows that such deep learning techniques can be successfully applied to heterogeneous data types, and in situations where the size of the training set is relatively limited (compared to the typical training sets used in other deep learning tasks such as image recognition).

Similar results have been shown for gesture recognition with wearable sensors using deep learning techniques by Duffner et al. [49] and Lefebvre et al. [89].

3.2.3 Hybrid activity recognition

Hybrid approaches combine both knowledge-driven and data-driven techniques, in an attempt to benefit from the strength of both while trying to overcome the drawbacks of one using the other. There are two intuitive ways to combine these two techniques: we can either improve a data-driven approach using expert knowledge and models, or we can improve the rules and model of an expert system using statistically extracted information from empirical data. We will refer to these two combination approaches as *knowledge-enhanced data-driven* approaches and *data-enhanced knowledge-driven* approaches respectively. We discuss in more details 3 papers that present such approaches in Section 3.2.3.1, Section 3.2.3.2, and Section 3.2.3.3.

3.2.3.1 Probabilistic logic programming for activity recognition

In [142], Sztyler et al. present an ADL recognition approach in homes using ProbLog, a probabilistic extension of Prolog. Prolog itself is a declarative logic programming language in which we express programs in terms of facts (which always succeed) and rules (which either succeed or fail, depending on its clauses). On the other hand, in ProbLog, facts and rules are assigned probabilities of success, which can be used to compute the probability of success of any query.

Sztyler et al. propose to use ProbLog to recognize ADL on fixed-sized windows of sensor events, by programming rules stating that certain sensor events correspond, with some probability, to certain activity classes. Querying the memberships of an activity instance to each activity class allows to assign that instance the activity class for which the membership has the highest probability of success.

However, unlike purely knowledge-driven approaches, the probabilities assigned to facts and clauses in their approach are computed from the frequencies of occurrence of sensor events during activity classes in a recorded training set. In that sense, their work is a data-enhanced knowledge-driven approach, which combines logical reasoning with data extracted from a training set. They show through this paper that such techniques are thus applicable to activity recognition in smart homes.

3.2.3.2 Learning situation models in homes

Brdiczka et al. present in [26] a framework for providing context-aware services in smart homes, based on learning situation models. They define situation models to be a set of entities (e.g. occupants in a home), the roles played by those entities (i.e. features extracted from sensor data, such as their posture or whether they are talking or not), and their relation with each other.

Their framework for providing context-aware services consists in 4 main steps: first, roles are extracted from sensor data; then, situations are segmented in an unsupervised fashion from the extracted roles; third, situations are labelled using supervised machine learning; finally, feedback of occupants on the services they expect to receive depending on the situation is collected. In this last step, occupants can provide preferences of expected services for situations that are too specific for the situation model learned in step 3 (e.g. asking for two different services when the occupant is using their computer for leisure and when the occupant is using it for work, but only the subsuming situation "Using the computer" was learned). In such cases, the subsuming situation is removed from the learned model and the subsumed situations for which the occupant gave feedback are learned in a supervised fashion.

In this approach, Brdiczka et al. thus propose a knowledge-enhanced data-driven approach for situation modelling (which includes activity recognition). In this work, however, additional knowledge used to improve data-driven techniques is provided not by actual experts, but by the occupants of the smart home, which give feedback on the services they expect and thus on the set of situation classes that must be modelled by the system. This paper thus shows that knowledge-enhanced data-driven approaches for activity recognition can benefit from knowledge provided not only by the designer of the smart home, which is the intuitive source of knowledge one first identifies, but also from occupants themselves. The knowledge they can provide will most likely be more specific to their particular home and needs than knowledge provided by experts, and

therefore most likely more valuable (although asking occupants for feedback in a non-intrusive and non-inconvenient manner is a problem in itself).

3.2.3.3 Activity recognition using visual localization

Wu et al. propose in [160] a vision-based technique for activity recognition in smart homes. Among the 3 methods they compare, one of them consists in using a different classifier for each camera used to record data. As such, each classifier will learn to recognize an activity always from the same viewpoint. In addition, the authors propose to limit the set of activities of each classifier to those that can actually be observed from that viewpoint (e.g. "Eating" can only be seen from the camera pointing at the dining table).

Recognizing activity then consists in selecting the right classifier (i.e. the right viewpoint) depending on which activity is really occurring. They propose to select the one in which there are the most extracted spatio-temporal features during the activity instance, that is intuitively, the viewpoint in which the most amount of visual changes occur.

This work presents a knowledge-enhanced data-driven approach for activity recognition in homes. Indeed, knowledge about the set of activity classes that can be observed by each camera is a necessity for it to be applied, which is assumed to be given by some expert. This requirement also implies that the models learned with this approach, as they argue in the paper, is not easily transferable from one home to the next, since the viewpoints, as well as the sets of observable activities, will change significantly from one camera to the next.

3.2.4 Discussion

We have seen in this section that both knowledge-driven approaches and data-driven approaches have been successfully applied to the problem of activity recognition in smart homes. However, these works also depict the limits of both approaches. In knowledge-driven approaches, designing rules and ontological representations of activity in the home can be very costly. Comprehensively capturing the specificities of each activity class that can occur in the myriad of different possible home configurations, sensor installations and occupants' routines also seems very difficult. In short, the number of expertly-designed rules required for a smart home system to be accurate at recognizing activities in a specific home will thus be very high [47].

As for data-driven approaches, most techniques currently rely on supervised machine learning, which typically requires large amounts of labelled data. Although this might not be as big of a problem in certain areas (e.g. object recognition in images, for which we can now find datasets of millions of examples), this is a very limiting factor in the case of smart homes. Indeed, if one intends to precisely learn the routines of a specific home, one needs to acquire labelled data for that home (and not learn a model from generic homes, which might

be moderately accurate for any home, but never very accurate for any home). Therefore, the system needs to record activities of occupants for a potentially long period of time (weeks, months, or even years) in order to gather sufficiently many activity instances to train the activity recognition approach [47]. During this time, the system is not operational and thus cannot provide any service based on activity information. In addition, an expert (which, in most current systems, is the occupant themselves) will have to label these instances with the corresponding activity label, which is inconvenient and error-prone especially if there are many instances.

Through our survey, we identified several approaches that we call hybrid, where both knowledge-driven and data-driven approaches are combined. These hybrid approaches seem to be promising for activity recognition in smart homes, as they represent a compromise between knowledge-driven and data-driven techniques. For example, we can hope that a knowledge-enhanced data-driven technique will require less training data than purely data-driven techniques, as well as requiring less knowledge than purely knowledge-based techniques, while maintaining good activity recognition performances.

Constraining context dimensions based on the activity, as done in [165] (presented in Section 3.2.1.2) or in [160] (presented in Section 3.2.3.3), is a promising idea as context dimensions often seem to be closely related. In particular, we can imagine using these context dimension constraints to simplify the model that a machine learning classifier has to learn. We present the motivations for our contributions on such a knowledge-enhanced data-driven approach in Section 3.3.1.

3.3 Place-based activity recognition

In the following section, we present our contributions to the problem of HAR from sensor data in smart homes, through what we call the *place-based* activity recognition approach. We first introduce in Section 3.3.1 the motivations that led to the construction of this approach. Then, we present the approach itself in Section 3.3.2. For each of the 3 main computing steps of the place-based approach, we present some classical algorithms that we will employ to evaluate the place-based approach in later experiments in Section 3.3.3 (for preprocessing), Section 3.3.4 (for classification), and Section 3.3.5 (for decision fusion). We conclude this section with a presentation of the results we expect to observe in experiments in Section 3.3.6.

3.3.1 Motivations

Activity is only one of four primary context dimensions in the home, as defined in Section 2.1.2.1. Consequently, one can wonder what relationships, if any, exist between these primary context dimensions of identity, time, place and activity. Dey and Abowd in [46] discuss the relationships between primary

and secondary context dimensions, and argue that secondary context is always indexable on primary context (we propose counter-examples to this in Chapter 5). However, they do not say anything about the relationships between elements of primary context themselves.

Intuitively, it seems that each of the 4 primary context dimensions are strongly related: for example, at night time (time dimension), occupants usually sleep (activity dimension); when an occupant is sleeping (activity dimension) in Bob's room (place dimension), that occupant is most likely Bob (identity dimension); etc.

In particular, the dimension of place seems to greatly influence the dimension of activity. For example, an occupant tends to associate the bathroom with a specific set of possible activities (showering, brushing their teeth, etc.); conversely, an occupant tends to associate the activity of taking a shower with the bathroom. This strong relationship between activities and places can be simply explained by the fact that most activities require interactions with specific parts and appliances of the home, and that these parts and appliances of the home are physically fixed in specific places. As such, activities are most often only physically performable in a subset of all places (and in particular, often in only one place).

As argued in Section 3.2, hybrid approaches are promising in smart home environments. We have seen in [160], presented in Section 3.2.3, that place information (the occupant's location in the home) can help improve activity recognition. Inspired by these works on hybrid approaches and previous observations on the link between place and activity, we seek to propose an activity recognition approach that exploits expert knowledge about place (the location of sensors, and the location of activity classes) to improve performances. However, contrary to the contributions in [160], we want to avoid designing specific rules to locate occupants in the home before activity recognition, as this is not easily generalizable to any home and adds an additional layer of possible errors.

3.3.2 The place-based approach

Let $\mathcal{S} = \{S_1, \ldots, S_n\}$ be the set of all sensors in the home, and let $\mathcal{A}$ be the set of all activity classes. As we have seen in Section 3.2.2, state-of-the-art data-driven approaches for activity recognition usually consist in a single classifier which must recognize the label of the current instance $a \in \mathcal{A}$ using exclusively data produced by all sensors of the home $\mathcal{S}$ (see Figure 3.2). We will call these approaches *global approaches* in the rest of this thesis.

Based on our observations made in Section 3.3.1, we conjecture that recognizing activities in a specific room, instead of the entire home, is a much simpler task. Let $\mathcal{S}^{(i)} = \{S_1^{(i)}, \ldots, S_{n_i}^{(i)}\}$ be the set of all sensors in the i^{th} place of the home, and let $\mathcal{A}^{(i)}$ be the set of activity classes which can occur in that place. A *local approach* for the i^{th} place consists in a classifier that seeks to recognize the label of the current instance $a \in \mathcal{A}^{(i)} \cup \{\text{none}\}$ using only data produced by sensors of that place $\mathcal{S}^{(i)}$. The addition of the dummy class "none" is necessary

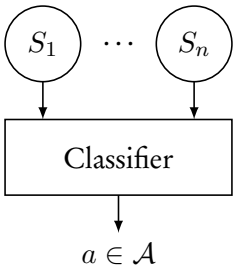

Figure 3.2 – Global activity recognition scheme.

(unless it is already part of $\mathcal{S}^{(i)}$), so that the classifier can assign a meaningful label to the instance, when said instance represents a situation where the occupant is performing an activity outside of the i^{th} place.

By learning local models for each of the places in the home, we could devise an activity recognition approach as follows: we first locate the occupant in the i^{th} place using a localization algorithm; we then recognize the activity of the occupant using the local model of the i^{th} place. This approach presents the disadvantage, as mentioned in Section 3.3.1, of requiring a localization algorithm, which not only adds complexity to the system, but also introduces a potential source of classification errors. Indeed, if the recognition system mistakenly locates the occupant in the wrong place, it almost surely will not properly recognize their activity, as it will use a set of input sensors that most likely did not capture any information related to that activity, and will only be able to choose among the classes of that place, which may not even contain the class of the real activity.

We instead propose a *place-based approach* where all local models are used simultaneously (see Figure 3.3). Let $\mathcal{P}$ be the set of all places in the home, and let $|\mathcal{P}|$ be the cardinal of that set. In our place-based approach, given an activity instance, local models 1 to $|\mathcal{P}|$ will first compute their respective sets of decisions $\{\Delta^{(1)}, \ldots, \Delta^{(|\mathcal{P}|)}\}$, where

$$\Delta^{(i)} = \{\delta^{(i)}_{1,1}, \ldots, \delta^{(i)}_{1,|\mathcal{A}^{(1)}|}, \ldots, \delta^{(i)}_{|\mathcal{P}|,1}, \ldots, \delta^{(i)}_{|\mathcal{P}|,|\mathcal{A}^{(|\mathcal{P}|)}|}, \delta^{(i)}_{\text{none}}\}, \tag{3.1}$$

and where $\delta^{(i)}_{k,j} \in [0, 1]$ is the decision of the classifier of the i^{th} place about the j^{th} activity class of the k^{th} place. In other words, each $\delta^{(i)}_{k,j}$ represents the degree of membership of the current activity instance to the j^{th} activity class of the k^{th} place, according to the classifier of the i^{th} place (in practice, many families of classifiers can output such membership degrees). In our thesis, since we decided that local models would only learn to recognize activity classes that can occur in their respective place, we have $\delta^{(i)}_{k,j} = 0$ if $i \neq k$ (extensions of this work could decide otherwise).

The goal of the decision fusion step is to compute the set of fused decisions $\overline{\Delta} = \{\overline{\delta}^{(1)}_1, \ldots, \overline{\delta}^{(1)}_{|\mathcal{A}^{(1)}|}, \ldots, \overline{\delta}^{(|\mathcal{P}|)}_1, \ldots, \overline{\delta}^{(|\mathcal{P}|)}_{|\mathcal{A}^{(|\mathcal{P}|)}|}, \overline{\delta}_{\text{none}}\}$ from the sets of decisions

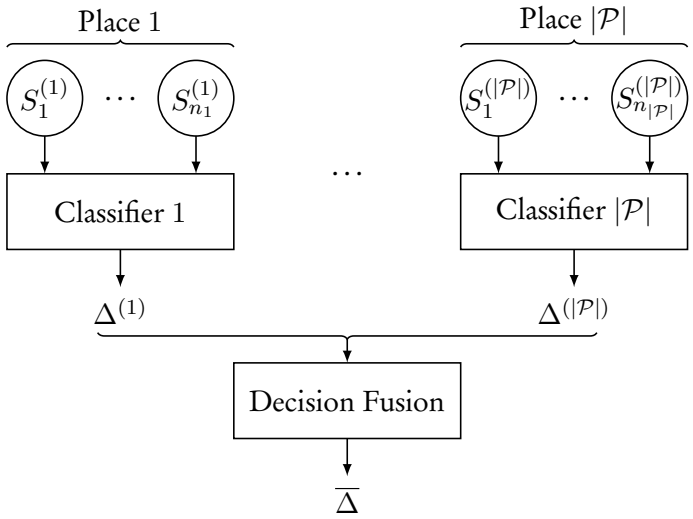

Figure 3.3 – Place-based activity recognition scheme.

$\{\Delta^{(1)}, \ldots, \Delta^{(|\mathcal{P}|)}\}$ taken in each place. From $\overline{\Delta}$, the smart home system can conclude that the class of the given instance is the a^{th} activity in the p^{th} place if that class has the maximum decision in $\overline{\Delta}$, that is:

$$\overline{\delta}_a^{(p)} = \max(\max_{j,k} \overline{\delta}_j^{(k)}, \overline{\delta}_{\text{none}}). \tag{3.2}$$

This decision assumes that the given instance corresponds to only one activity, as we had assumed in Section 3.1. If we assume that activities can occur simultaneously throughout the home, we can modify this decision process such that multiple activity labels are given to the instance, using the maximum values in $\overline{\Delta}$.

Special considerations must be taken for class "none". Indeed, when an activity occurs in a certain place, we expect all classifiers of other places to decide in favour of activity "none", as they are not impacted by the activity. Therefore, we must make sure that activity "none" is only decided when all classifiers agree that it is the most likely label. Before deciding on the label of the instance using Equation 3.2, we thus compute the value of $\overline{\delta}_{\text{none}}$ as such:

$$\overline{\delta}_{\text{none}} = \begin{cases} 0 & \text{if } \exists\, i \in \mathcal{P}, \exists\, (j,k) \in \mathcal{A}^{(i)} \times \mathcal{P}, \delta_{k,j}^{(i)} \geqslant \delta_{i,\text{none}} \\ 1 & \text{otherwise} \end{cases}. \tag{3.3}$$

We discuss in more details each of the place-based approach steps in later sections: in Section 3.3.3, we discuss the preprocessing necessary before each local classification; in Section 3.3.4, we present some of the classical families of classifiers that can be used to perform local classification; in Section 3.3.5, we present some of the classical decision fusion algorithms that we can employ in our place-based approach. However, we first discuss in Subsections 3.3.2.1 to 3.3.2.6

some of the characteristics of our place-based approach, as compared to global approaches.

3.3.2.1 Required knowledge

Our place-based approach is a knowledge-enhanced data-driven approach, like the ones presented in Section 3.2.3, in the sense that some expert knowledge is required to set up the process: the membership of each sensor to places of the home, as well as the membership of each activity class to places of the home must be known. This is a downside of the approach, compared to a global approach that does not require this knowledge. However, some further work (not addressed in this thesis) could help reduce the cost of acquiring this knowledge.

Sensors installed in the same place could be discovered automatically using clustering methods. Indeed, if sensors are spatially close to each other, they will most likely capture data about the same situations, and we will thus find correlations in the data they collected. Such clusters of sensors could constitute a basis for the memberships of sensors to places. Those memberships could also be given in the future by home contractors, assuming that future homes will be instrumented immediately upon construction. A present-day solution would be to ask the user of the smart home system to indicate the location of each sensor they installed (which is possible to do in many current smart home solutions).

As for activities, similar unsupervised methods could be used to try to discover groups of activity classes among collected data. Such methods are starting to be studied, as in the works of Cook et al. in [33]. Membership of activity classes to places could also be set by home experts from a general model of activities in home (for example, by setting that activity "Showering" can only occur in a bathroom). Contrary to the works of Wu et al. [160] presented in Section 3.2.3.3, each model in our place-based approach is associated to a specific place of the home, as opposed to a viewpoint; many places are similar from one home to the next, in the sense that the set of activities we can perform in the place of one home will be similar to the set of activities of the corresponding place in another home (e.g. cooking-related activities in kitchen-like places). Establishing such sets of activity classes in places using expert modelling is thus realistically possible, whereas viewpoint-based approaches cannot benefit from such expertise, as the set of activities solely rely on the viewpoint of the camera, which can be vastly different from one home to the next, even if they monitor corresponding places.

3.3.2.2 Localization through place-base activity recognition

Our place-based approach does not need to locate the occupant before recognizing activities. Quite the contrary, it actually locates the occupant as well as recognize their activity, since each activity class is linked to the place in which it can occur. Later work (not addressed in this thesis) could investigate the use of place-based activity recognition to improve localization algorithms, or conversely

the use of localization algorithms to improve place-based activity recognition, or even both in conjunction to improve the global performances of the system.

3.3.2.3 Agnosticism to sensor types, classifier types, and decision fusion algorithms

Our place-based approach is agnostic to the types of sensors installed, in the home, the classifiers used in each local model, and the decision fusion algorithm used to combine local models. Any combination of sensors can be used as data sources for a place, as long as the preprocessing step and the classifier of that place are tailored to process this data (for example, special considerations must be taken into account when input data are videos). Any classifier type can be used in a local model, so long as it can output the set of decisions $\Delta^{(i)}$, which is possible for the vast majority of classical classifier types. Similarly, any decision fusion method can be used.

This agnosticism is an appreciable benefit of the place-based approach. The rapid growth of connected objects and sensor capabilities will thus not render our place-based approach quickly obsolete, but rather always up-to-date in terms of data collection. Similarly, the extensive recent and future developments in classification algorithms and decision fusion techniques will always be integrable in a place-based scheme. On the other hand, a specific global classifier for activity recognition in smart homes tailored to certain data types will not be able to integrate future developments on sensor and classification technologies as seamlessly.

3.3.2.4 Multi-classifier fusion

A direct consequence of the agnosticism of the place-based approach to the types of classifiers used is that we do not have to restrict each place to use only one classifier. Since we introduced decision fusion to combine local models together, we can also use it to combine multiple classifiers in each local model (which is the classical purpose of decision fusion). Similarly, we can combine multiple classifiers through decision fusion in a global approach.

Combining multiple classifiers can potentially allow better activity recognition performances, with very limited computing overhead, assuming these classifiers make different classification mistakes and thus can assist one another. We will refer to this approach as *multi-classifier fusion* in the experimental sections.

3.3.2.5 Non-monolithicity

Our place-based approach is non-monolithic in the sense that the local model of a place is independent of all the other local models. As such, each local model can be trained and executed independently from all other models, and any modification to one of the local models is guaranteed to not have any effect on any of the other local models.

This modularity of the approach presents obvious benefits in terms of execution: the place-based approach can be easily parallelized by spreading the training or run-time execution of each local model to different computing cores. This should lead to great computing time improvements, if decision fusion is not a long operation (which it often is not).

Moreover, this modularity is advantageous when the home environment is changing. Since the training phase of each local model is independent, we can retrain one local model when changes to the place it monitors occur. For example, if the occupants bought and installed a new sensor in a particular place, we can retrain the model of that place to take this new source of data into account, without changing any other model. More generally, we can retrain each model independently from the others as time passes and routines of occupants change.

3.3.2.6 Applicability to multi-occupants scenarios

Possible perspectives on the place-based approach (not addressed in this thesis) include the recognition of simultaneous activities of different occupants in the home. Using our place-based approach, one could modify the decision step so that it takes into account the number of occupants present in the home; it then could recognize multiple simultaneous activities performed by these occupants, so long as they are located in different places. This would only require changes to the decision fusion step and not to any other part of the approach.

When using global approaches, this is much less direct of a change. Indeed, naïvely, a global approach would then need to recognize not activities, but rather sets of activities. The number of possible sets of activities from $\mathcal{A}$ is $2^{|\mathcal{A}|}$, which grows exponentially with the number of activity classes. With the place-based approach, each local model will still recognize activities, and thus the complexity of the approach is by construction equivalent to mono-occupant situations.

3.3.3 Preprocessing

As our proposed place-based approach is agnostic to the type of classifier used in each place, it is essential to process raw sensor data before the classification step, in a way that is well-adapted to the classifier used in each place. We present in this section 3 preprocessing steps that are often required when the sources of data are ambient smart home sensors or wearable sensors, and when we use classic classifiers such as those presented in Section 3.3.4: filling missing values (presented in Section 3.3.3.1), normalizing data (presented in Section 3.3.3.2), and reducing noise (presented in Section 3.3.3.3).

In the following subsections, we will denote by s_t the value of a sensor at timestep $t \in [\![0, T]\!] = \{0, 1, \ldots, T-1, T\}$.

3.3.3.1 Missing values

Many smart home sensors are autonomous and thus send their data using wireless protocols. In order to reduce battery usage, these smart home protocols often do not include acknowledgement mechanisms. As such, it is not uncommon that some of the frames these sensors send get lost. In particular, some of the datasets we find in the literature do have a number of missing values, which in general cannot be processed by classifiers. As such, a preprocessing step is needed to fill in those missing values. We present in this section 3 classical interpolation approaches. We present on Figure 3.4 a visual comparison of those 3 approaches.

Let $(i, j) \in [\![0, T]\!]^2, i < j$, be two timesteps between which all values $s_k, i < k < j$, are missing values.

Last observation carried forward A series of missing values is replaced with the last value that was not missing:

$$\forall k \in [\![i + 1, j - 1]\!], s_k = s_i. \tag{3.4}$$

Linear interpolation A series of missing values is replaced with values computed from an interpolated line that passes through v_i and v_j:

$$\forall k \in [\![i + 1, j - 1]\!], s_k = s_i + (k - i)\frac{s_j - s_i}{j - i}. \tag{3.5}$$

Spline interpolation A series of missing values is replaced with values computed from interpolated polynomial segments. Cubic splines are some of the most common splines used for this process [40].

3.3.3.2 Normalization

The range of values sensors can provide in the home can greatly vary, depending on their type, the environment, or even their calibration. However, such varying ranges can have a negative impact on the performances of many classifiers. Indeed, classifiers may give more importance to sensors which provide high absolute values, compared to those that provide values close to 0, even though they should have equal importance. A normalization step can erase this problem by updating the values of each sensor such that all sensor ranges are identical. In this section, we present 2 of the main ways of computing a normalized value s'_t from a value s_t.

Rescaling We can rescale data from a sensor such that each data point falls into an interval $[a, b]$ (typically, $[0, 1]$ or $[-1, 1]$). Let s_{inf} be the infimum and s_{sup} the supremum of the set of values of s; we have:

$$s'_t = a + \frac{(s_t - s_{\text{inf}})(b - a)}{s_{\text{sup}} - s_{\text{inf}}}. \tag{3.6}$$

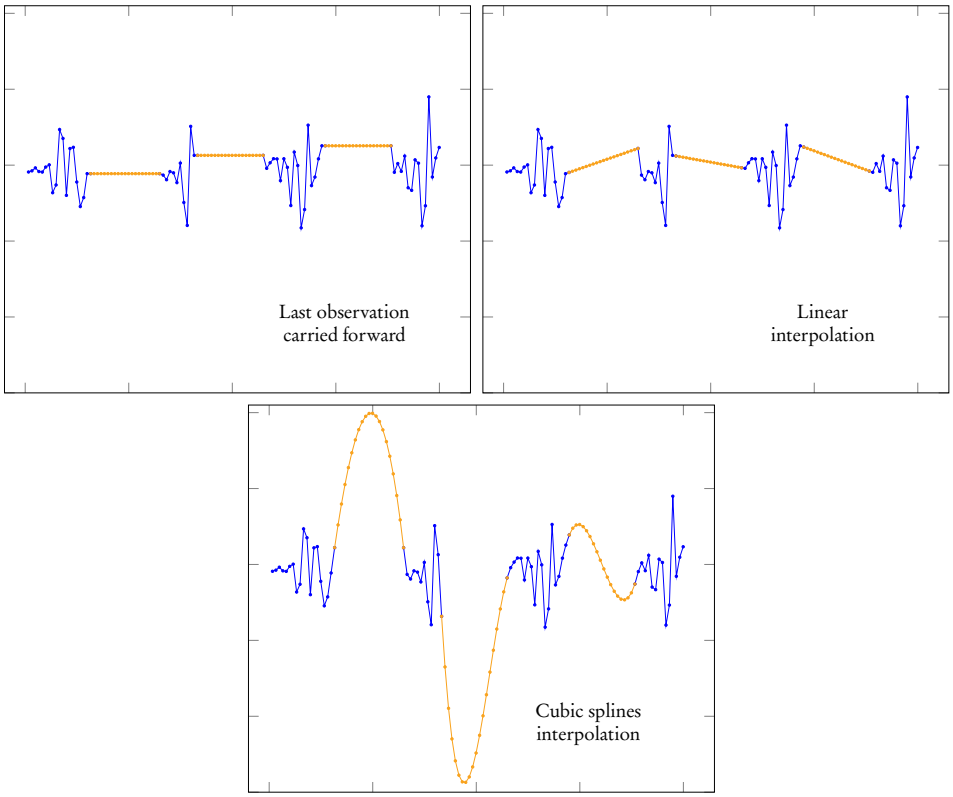

Figure 3.4 – Filling missing data using one of the 3 interpolation methods presented in Section 3.3.3.1. Known data is represented in dark blue, and interpolated data is represented in light orange.

The two bounds s_{inf} and s_{sup} can usually be known in advance (e.g. an accelerometer is calibrated such that its output value is always in $[-3, 3]$). However, it is not uncommon that those bounds are not known for a sensor, or that one of the two bounds or both is not finite. In such cases, we can typically estimate s_{inf} and s_{sup} from the empirically smallest and biggest values found in the training data; in that case, this means that on new data, the rescaling process can result in values outside the interval $[a, b]$.

Standardization We can standardize data from a sensor such that they have a mean of 0 and a variance of 1. Let $\overline{s}$ be the mean and σ the standard deviation of the values outputted by the sensor in the training dataset; we have:

$$s'_t = \frac{s_t - \overline{s}}{\sigma}. \tag{3.7}$$

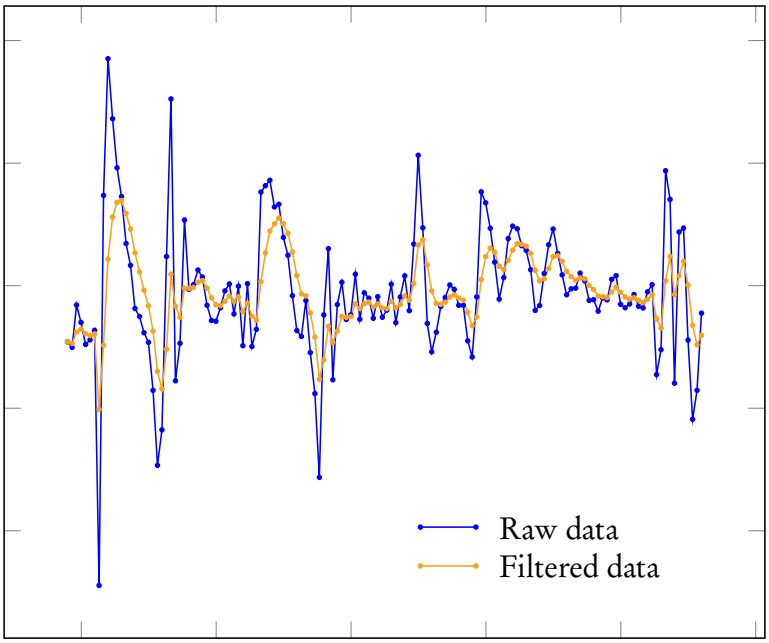

Figure 3.5 – Filtering data using the basic approach presented in Section 3.3.3.3, with $\beta = 0.3$.

3.3.3.3 Noise reduction

Raw sensor data can often be noisy; in particular, wearable sensors such as accelerometers or gyrometers can pick up micro-variations in gestures that are not useful to characterize activities. Quite the contrary: such micro-variations can be learned by the classifier during training to recognize certain activity instance, which will degrade its generalization performances on unseen data.

There are many different techniques to reduce noise, depending on the type of data at hand (e.g. specific denoising processes exist for images). An extensive presentation of noise reduction approaches is presented in [154]. In our work , we will use a basic filtering method, controlled by a parameter $\beta \in [0, 1]$. We filter sensor data as such:

$$s'_t = \beta s_t + (1 - \beta) s_{t-1}. \tag{3.8}$$

The case $\beta = 0$ corresponds to maximum filtering, where $s'_t = s_0$ for any t. The case $\beta = 1$ corresponds to minimum filtering, where $s'_t = s_t$ for any t. Figure 3.5 presents the result of applying this noise reduction method with $\beta = 0.3$ (in light orange) on noisy data (in dark blue). We can see that this method mostly conserves the shape of the original signal while softening noisy oscillations.

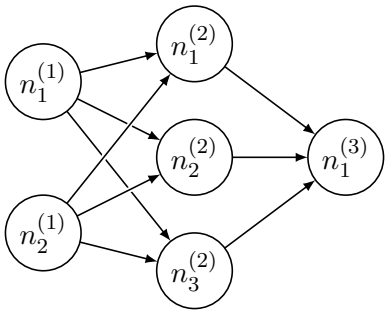

Figure 3.6 – Example of a MLP with an input layer of 2 neurons, one hidden layer of 3 neurons, and an output layer of 1 neuron.

3.3.3.4 Conclusion on preprocessing

The usefulness of preprocessing in our place-based approach will highly depend on the sensors installed in the home as well as the classifiers used in each place. Some classification techniques such as deep learning methods argue that there should be as little preprocessing as possible applied to data, because deep learning classifiers should learn to extract relevant features directly from raw data.

In our thesis, our focus is not on a specific classifier design but rather on a classification architecture well-fitted to smart home systems. As such, we will use classical classifiers, which typically require more preprocessing than newer methods. We present these classifiers in the next section.

3.3.4 Classification

We present in this section some of the classical classifiers of the literature that can be employed in our place-based approach, as our approach is agnostic to the types of classifier used. We present, in each subsection, one archetype of common classifier families: MultiLayer Perceptrons (MLPs) for ANN in Section 3.3.4.1, SVMs for kernel methods in Section 3.3.4.2, Bayesian Networks (BNs) for probabilistic graphical models in Section 3.3.4.3, and Dynamic Time Warping (DTW) for geometric similarity measures in Section 3.3.4.4.

Other classification techniques such as decision trees, Hidden Markov Models (HMMs), conditional random fields, etc., have been employed in the literature and could thus be used in our place-based approach as well (not addressed in this thesis).

3.3.4.1 Multilayer perceptrons

A MLP is a feedforward ANN (that is, with no loops) in which the output of a neuron is fully connected to the input of all the neurons in the next layer. We give a graphical representation of an example MLP on Figure 3.6.

Let $n_i^{(j)}$ be the i^{th} neuron in layer j, and let $y_i^{(j)}$ be the output of this neuron. The inputs of $n_i^{(j)}$ is the set of outputs $\{y_1^{(j-1)}, \ldots, y_{N_{j-1}}^{(j-1)}\}$ of the N_{j-1} neurons of the previous layer $j-1$. We can then compute $y_i^{(j)}$ as follows:

$$y_i^{(j)} = \varphi\left(b_i^{(j)} + \sum_{k=1}^{N_{j-1}} w_{k,i}^{(j-1)} y_k^{(j-1)}\right), \tag{3.9}$$

where $w_{k,i}^{(j-1)}$ is the weight between neuron k of layer $j-1$ and neuron i of layer j, where $b_i^{(j)}$ is the bias of neuron i of layer j, and where φ is the activation function of the neuron. This activation function is typically a sigmoid function such as the logistic function $x \mapsto \frac{1}{1+e^{-x}}$ or the hyperbolic tangent function $x \mapsto \frac{1-e^{-2x}}{1+e^{-2x}}$ [21].

To associate a class to an input, we set the number of output neurons to match the number of classes [1]: the recognized class is then the one corresponding to the output neuron with the biggest output. The set of these outputs will be the decisions set Δ used in our place-based approach.

The weights and biases of the entire network can then be learned in a supervised training step, using the gradient backpropagation algorithm [129] or its many variants. On the other hand, the topology of the network (number of layers, number of neurons per layer) is generally set by hand. A number of heuristics for training ANN efficiently are presented in [86].

Cybenko in [38] proved the universal approximation theorem: an MLP with one hidden layer of finitely many neurons, using sigmoidal activation function [2], can approximate any continuous function on compact subsets of $\mathbb{R}^n$. However, this theorem says nothing about how to find the parameters of the MLP to approximate a specific function.

3.3.4.2 Support vector machines

SVMs, first introduced in [23], are a generalization of linear classifiers for 2-classes problems.

Suppose we have a vector of input data $\boldsymbol{x} = (x_1, \ldots, x_N)$. We can construct a linear classifier using a vector of weights $\boldsymbol{w} = (w_1, \ldots, w_N)$:

$$y = \boldsymbol{w}\boldsymbol{x}^\intercal + w_0. \tag{3.10}$$

The instance will be classified in class 1 if $y \geqslant 1$, and in class 2 if $y \leqslant -1$.

The goal is then to learn the separating hyperplane $y = 0$ using the training set (that is, examples of $\boldsymbol{x}$ associated with the class label). Since the training set is

1. Except in 2-classes problems where only one neuron can be used: if the output of that neuron is in $[0, 1]$, an output closer to 0 will correspond to the first class while an output closer to 1 will correspond to the second class.
2. Later work by Hornik [68] showed that other functions can be used, under mild assumptions.

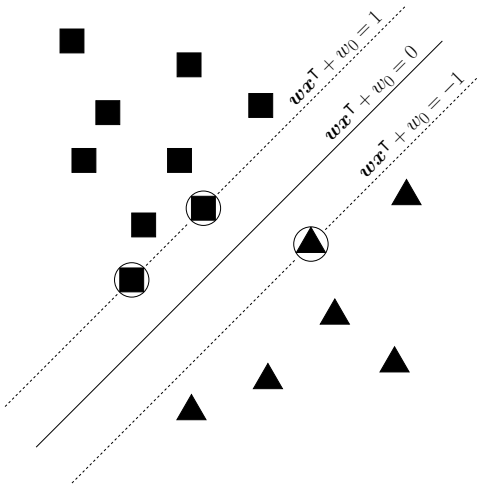

Figure 3.7 – Maximum margin hyperplane on a 2-dimensional training set of 2 classes (squares and triangles), in a linearly separable case. The support vectors are circled.

finite, there are infinitely many such hyperplanes. In SVMs, the hyperplane that maximizes the margin between the two classes in the training set is chosen, by minimizing the norm of $\boldsymbol{w}$ while respecting the constraints of memberships of the training examples to their classes. The training samples that lie on the margin are called the *support vectors*. We give an example of such a separating hyperplane on Figure 3.7.

Minimizing $\|\boldsymbol{w}\|^2$ under those constraints can be solved through a variety of optimization algorithms [21]. In practice, we often use soft margins rather than hard margins: we allow that some training samples are on the wrong side of the margin, compared to their labels. In this case, we seek to minimize $\|\boldsymbol{w}\|^2 + C\sum_{p=1}^{P}\max(0, 1 - y_p(\boldsymbol{w}\boldsymbol{x}^\intercal + w_0))$ instead; when all training samples are on the right side of the margin, the sum is equal to 0 and the problem is correctly identical to hard-margin SVMs. Constant $C > 0$ allows to control the compromise between the number of training samples on the wrong side of the margin and the width of the margin itself. Soft-margin SVMs allow to work with data that is not quite linearly separable; in particular, it can allow better generalization performances in cases where noise in the data make the theoretical general margin difficult to find.

In most real-world classification problems, classes are not linearly separable at all. In such cases, SVMs make use of the *kernel trick*: we transform the data from its original space into a space of higher dimension, in which it is more probable that the problem is linearly separable. More precisely, we now define the output label as follows:

$$y = \boldsymbol{w}\boldsymbol{\varphi}(\boldsymbol{x})^\intercal + w_0, \tag{3.11}$$

where $\boldsymbol{\varphi}$ is a non-linear transformation that verifies a number of properties

described in [21]. We can then find the maximum margin hyperplane using the same techniques as in the linear case.

It is possible to adapt the standard SVM approach to solve multi-class problems. For example, one can construct as many SVMs as the number of classes, where each of these SVMs will be trained to classify one class against all other classes; we can then combine them in order to get a multi-class classification algorithm that provides a set Δ of decisions for each class.

3.3.4.3 Bayesian networks

A BN is a directed acyclic graph which represents a joint probability distribution of a set of variables. For example, in Figure 3.8, we model a BN with 5 variables $\{x_1, \ldots, x_5\}$, where the directed edges represent conditional dependencies. The joint probability distribution of this example BN is then:

$$\mathrm{p}(x_1)\mathrm{p}(x_2 \mid x_1)\mathrm{p}(x_3 \mid x_1)\mathrm{p}(x_4)\mathrm{p}(x_5 \mid x_2, x_3, x_4). \tag{3.12}$$

In an activity classification task, one of the variables of the BN is the activity class (say, x_1 in our example) and the other variables represent sensor data ($\{x_2, x_3, x_4, x_5\}$ in our example). Classifying an instance, given specific sensor values for x_1, x_2, x_3, and x_4, consists in computing:

$$\underset{x_1}{\operatorname{argmax}}\ \mathrm{p}(x_1 \mid x_2, x_3, x_4, x_5). \tag{3.13}$$

In our place-based approach, the set of decisions Δ of the BN is the set of $\mathrm{p}(x_1 \mid x_2, x_3, x_4, x_5)$ for each value of x_1 (that is, each possible class label).

The structure of the BN as well as the conditional probabilities of the BN can be learned from training data, through various approaches [25]. For example, measuring conditional independence between two variables in the training set can imply that there is no edge between those two variables in the BN structure. Structure learning is a more difficult problem than distribution learning [16].

3.3.4.4 Dynamic time warping

DTW is a geometric approach for comparing 2 time-dependent sequences of values [135]. Let x_t and $y_{t'}$ be the values of two sensors at timesteps $t \in [\![1, T]\!]$ and $t' \in [\![1, T']\!]$. In order to compare x_t and $y_{t'}$, we need a cost measure $c : [\![1, T]\!] \times [\![1, T']\!] \longrightarrow [0, +\infty)$; this cost function must output small values when x_t and $y_{t'}$ are "similar", and large values if they are "dissimilar" (a geometric example would be the euclidean distance). We can then compute a cost matrix that contains the values of c when applied to each couple $(x_t, y_{t'}) \in [\![1, T]\!] \times [\![1, T']\!]$. We present on Figure 3.9 an example of a cost matrix between two sequences where c is the Manhattan distance.

We can now align sequences together by walking through the cost matrix, from the start of the sequences $(0, 0)$ to the end of the sequences (T, T'). More

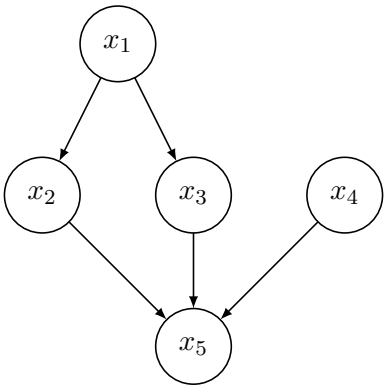

Figure 3.8 – A bayesian network of 5 variables. The directed edges represent conditional dependences.

precisely, we can define a warping path as a sequence $p = (p_1, \ldots, p_n)$, where $p_i = (t_i, t'_i) \in [\![1, T]\!] \times [\![1, T']\!]$. A warping path must respect two conditions:

— boundary condition: $p_1 = (1, 1)$ and $p_n = (T, T')$;

— step size condition: $\forall i \in [\![1, n-1]\!], p_{i+1} - p_i \in \{(1, 0), (0, 1), (1, 1)\}$.

The boundary condition guarantees that the start and end points of both sequences are necessarily aligned together. The step size condition enforces that all elements of both sequences are part of the warping path at least once and that there are no duplicate pairs in p; it also enforces the monotonicity of the path, that is that $t_i \leqslant t_{i+1}$ and $t'_i \leqslant t'_{i+1}$.

The cost c_p of a warping path p is the sum of the costs we get when following the path through the cost matrix:

$$c_p(x, y) = \sum_{i=1}^{n} c(x_{t_i}, y_{t'_i}). \tag{3.14}$$

The DTW distance between two sequences is then the cost of the optimal warping path, that is the smallest possible cost among all warping paths. For example, on Figure 3.9, the white line is the optimal warping paths for those two sequences using the Manhattan distance. Finding this minimum cost is an optimization problem which can be solved using a variety of techniques [135].

DTW can be applied to activity recognition as follows: we evaluate the distance between the input instance to a training instance by summing the costs of the optimal warping paths of each sensor. We can then decide what the label of the input instance is based on which training instances are the closest to that input instance, from the previously computed distances. For example, the input instance can be assigned the label of the closest training instance; it could also be assigned the label of the class, for which the average of all distances between the input and training instances of that class is the smallest. The set of these distances to each class can be thought as the set of decisions Δ in our place-based approach.

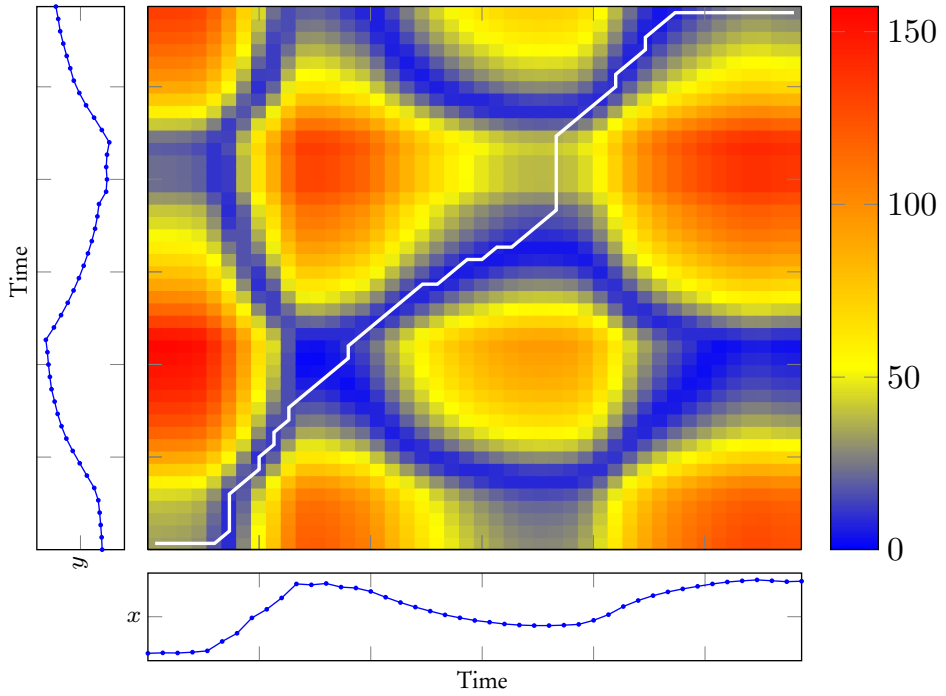

Figure 3.9 – Cost matrix of two sequences (pictured on the left and below the matrix) using the Manhattan distance (absolute value of the difference). Low cost is represented in blue while high cost is represented in red. The white line is the optimal warping path.

3.3.4.5 Conclusion on classification

As our place-based approach is agnostic to the type of classifier used in each place, we will experimentally study the performances of the place-based approach using each of the 4 classifier types presented in this section. Since those 4 classifiers are significantly different in the way they model activities from sensor data, we hope to show that the place-based approach performs significantly better than global approaches with any of these classifier types.

Similarly to classifiers, we present in the next section some classical decision fusion methods we can use in our agnostic place-based approach.

3.3.5 Decision fusion

We present in this section some decision fusion techniques of the literature that can be used in our place-based approach. We present in voting methods in Section 3.3.5.1, stacking methods in Section 3.3.5.2, and probabilistic methods with Dempster-Shafer Theory (DST) in Section 3.3.5.3 and possibility theory in Section 3.3.5.4.

3.3.5.1 Voting

In voting methods, the classifier of each place i gives a vote $v_{k,j}^{(i)} \in [0,1]$ about the j^{th} activity of the k^{th} place. The fused decision $\overline{\delta}_j^{(k)}$ about the j^{th} activity of the k^{th} place is then:

$$\overline{\delta}_j^{(k)} = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} v_{k,j}^{(i)}. \tag{3.15}$$

Factor $1/|\mathcal{P}|$ is here to ensure that $\overline{\delta}_j^{(k)} \in [0,1]$.

We can devise different voting schemes depending on how $v_{k,j}^{(i)}$ for all j and k are computed from $\Delta^{(i)}$. The most general scheme, called *weighted* voting, computes the votes using a weighting function $\varphi : [0,1] \rightarrow [0,1]$ such that $v_{k,j}^{(i)} = \varphi(\delta_{k,j}^{(i)})$. A common voting scheme is that of *majority* voting, where each classifier of place i gives a vote of 1 for the class it is most confident in, and 0 to all other classes:

$$v_{k,j}^{(i)} = \begin{cases} 1 & \text{if } \delta_{k,j}^{(i)} = \max\limits_{l,m} \delta_{m,l}^{(i)} \\ 0 & \text{otherwise} \end{cases}. \tag{3.16}$$

3.3.5.2 Stacking

Stacking methods, first introduced by Wolpert in [159], consist in solving the problem of decision fusion as a classification problem. Indeed, we can view decision fusion as a classification task where input data are the decisions $\{\Delta^{(1)}, \ldots, \Delta^{(|\mathcal{P}|)}\}$ from the classifiers of each place, and where the target output is the set of fused decisions $\overline{\Delta}$. We can then use any classification methods that can work with such inputs and target outputs; MLPs, or SVMs, are examples of such classifiers. These classifiers can be trained using the same training set that is used to train place-based classifiers, optimized using the same validation set, and tested using the same test set.

3.3.5.3 Dempster-Shafer theory

DST, also called evidence theory, is a theoretical frame that aims at modelling uncertainty and imprecision in data. DST emerged from the works of Dempster [43], which were continued by Shafer [136].

Let $2^{\mathcal{A}} = \{\varnothing, \{a_1^{(1)}\}, \{a_2^{(1)}\}, \{a_1^{(1)}, a_2^{(1)}\}, \ldots, \mathcal{A}\}$ be the power set of activity classes $\mathcal{A}$. The *mass function* m_i, associated to the classifier of the i^{th} place, is defined as:

$$m_i : 2^{\mathcal{A}} \longrightarrow [0,1], \tag{3.17}$$

$$\sum_{a_j^{(k)} \in 2^{\mathcal{A}}} m_i(a_j^{(k)}) = 1. \tag{3.18}$$

Choosing the right mass function is not an obvious task. For our purposes, we will only assign non-zero masses to singleton classes, as being the decision of the classifier for that place, normalized so that Equation 3.18 is respected:

$$\forall a \in 2^{\mathcal{A}}, m_i(a) = \begin{cases} \frac{\delta_{k,j}^{(i)}}{\sum_{l \in \mathcal{P}} \delta_{k,j}^{(l)}} & \text{if } a = \{a_{k,j}^{(i)}\} \\ 0 & \text{otherwise} \end{cases}. \tag{3.19}$$

This choice alleviates the need for devising more elaborate mass evaluation heuristics; it also greatly reduces the computing times of the next steps in DST decision fusion, as most images of elements of $2^{\mathcal{A}}$ through m_i are 0.

We can then combine the mass functions of all classifiers into a single mass function m. Smets' rule [139] is one example of such combination methods:

$$\forall a \in 2^{\mathcal{A}}, m(a) = \sum_{b_1 \cap \cdots \cap b_{|\mathcal{P}|} = a} \prod_{i \in \mathcal{P}} m_i(b_i). \tag{3.20}$$

The fused decisions are then computed using the *plausibility* function Pl:

$$\overline{\delta}_j^{(k)} = \text{Pl}(\{a_j^{(k)}\}) = \sum_{b \in 2^{\mathcal{A}}, a_j^{(k)} \in b} m(b). \tag{3.21}$$

3.3.5.4 Possibility theory

Much like DST, possibility theory also aims at modelling uncertainty and imprecision in data. Possibility theory, mostly developed by Dubois and Prade [48], is founded on Zadeh's theory of fuzzy sets [167].

Let $\mu_{k,j}^{(i)}$ be the degree of membership of the current instance to the j^{th} activity class of the k^{th} place, according to the classifier of the i^{th} place. Similarly to mass functions estimation in DST, choosing the right heuristics to compute this membership degree is difficult. In this work, we directly assign classifier decisions to membership degrees, i.e. $\mu_{k,j}^{(i)} = \delta_{k,j}^{(i)}$. We can then combine the membership degrees of each classifier into a single set of membership degrees which will be our fused decisions $\overline{\Delta}$, using a variety of different methods. We use here the formula proposed in [52]:

$$\overline{\delta}_j^{(k)} = \sqrt{\frac{\sum_{i \in \mathcal{P}} (w_i \cdot \mu_{k,j}^{(i)})^2}{|\mathcal{P}|}}, \tag{3.22}$$

where

$$w_i = \frac{\sum_{p \in \mathcal{P}, p \neq i} H_{0.5}(p)}{(|\mathcal{P}| - 1) \sum_{p \in \mathcal{P}} H_{0.5}(p)}, \tag{3.23}$$

and where

$$H_\alpha(i) = \frac{1}{2^{-2\alpha}} \sum_{k \in \mathcal{P}} \sum_{j \in \mathcal{A}_k} (\mu_{k,j}^{(i)})^\alpha (1 - \mu_{k,j}^{(i)})^\alpha. \tag{3.24}$$

3.3.5.5 Conclusion on decision fusion

We have presented some of the main decision fusion approaches used in the literature. As our place-based approach is agnostic to the decision fusion method used, we will employ each of these methods in our experiments.

In a multi-classifier place-based approach, we expect decision fusion to be only useful if classifiers make significantly different classification errors, in which case they can complement each other. If the classifiers of a same place have identical behaviour, decision fusion will only have an averaging effect, but will most likely not improve recognition performances.

3.3.6 Expected results

Following the presentation of our place-based approach, we can expect to obtain a number of results when performing activity recognition experiments on smart home datasets. In particular, we expect 2 main categories of improvements: those that are related to the actual recognition performances, and those that are related to the computing times of the approach.

3.3.6.1 Expected results on recognition performances

Our place-based approach follows the *divide and conquer* approach to problem solving, by dividing the modelling task of recognizing activities from sensor data among multiple classifiers, based on the location of those activity classes and sensors. Using a global approach, the more sensors and activity classes there are, the more complex it is for that approach to discover the right relationships between sensors and activity classes, using limited labelled data. On the other hand, with our place-based approach, this increase in complexity should be relatively limited, assuming that the increase in sensors and activity classes is well-distributed among places of the home. Indeed, the complexity of the model is split between all places of the home and the decision fusion step; individually, each model should be relatively simple to learn from limited labelled data.

Therefore, we first conjecture the following:

Hypothesis 3.1. *A place-based approach will on average achieve higher activity recognition performances than a global approach, for any classifier type used.*

The high variability in home layouts, sensor installations, and activity habits of occupants lead us to believe that there will also be a high variability in the type of classifier that works best to model a place of a home or an entire home. Multi-classifier fusion allows to circumvent the need for selecting the best type of classifier beforehand, and instead uses multiple classifiers for each model.

Therefore, we also conjecture the following:

Hypothesis 3.2. *Combining multiple classifier types through decision fusion leads to better recognition performances, for both place-based and global approaches.*

In order to circumvent the need for a preliminary estimation of the location of an occupant in the home, each place in our place-based approach should have the possibility of recognizing that nothing is happening, in addition to the other activity classes of said place. We thus introduce, in each place, an additional class named "None" that corresponds to this idle state of a place. However, if some sensors are shared between places (for example, sensors worn by the occupant), confusions might arise because sensors changes will be observed by multiple places.

Therefore, we finally conjecture the following:

Hypothesis 3.3. *Class "None", which corresponds to the absence of any activity, will be difficult to model by a place-based approach.*

3.3.6.2 Expected results on computing times

In much the same way that divide and conquer strategies lead to efficient sorting algorithms (e.g. merge sort [77]), we can expect our proposed place-based approach to have favourable behaviour in terms of computing times, compared to a global approach. In particular, since modelling is split between the different places of the home, we can expect that each place will individually be much simpler to model, and thus much faster to learn. In addition, we can assume that the decision fusion step will always be negligibly fast to execute, since its number of inputs will be small for most realistic homes.

Therefore, we can conjecture the following:

Hypothesis 3.4. *It will take less time to effectively train a place-based activity recognition approach compared to a global approach on a given dataset.*

For similar reasons, we can expect that activity recognition at run time will also be faster in the place-based approach. However, we expect the gap in speeds between a place-based approach and a global approach to be smaller in this case compared to training times. Indeed, machine learning techniques for which training times increase immensely as the complexity of the problem increases (e.g. neural networks) typically do not slow down by a large factor because of this complexity during run time.

Nevertheless, we conjecture the following:

Hypothesis 3.5. *It will take less time to recognize an activity at run time for a place-based approach compared to a global approach.*

3.4 Experiments

In order to empirically analyse the behaviour of the place-based approach we proposed in Section 3.3, we have performed a number of experimental evaluations, using 2 of the datasets presented in Section 2.3: the Opportunity dataset and the Orange4Home dataset. The first set of experiments we performed, as reported in Section 3.4.1, shed light on the recognition performances of our approach and thus on Hypotheses 3.1, 3.2, and 3.3. The second set of experiments we performed is related to the computing times of the place-based approach, so as to empirically validate Hypotheses 3.4 and 3.5.

In the following experiments, data were preprocessed as follows: missing values were replaced using cubic spline interpolation. A low-pass filter with $\beta = 0.1$ was applied to the data in order to reduce the influence of noise. Data were standardized such that the mean value of each sensor was 0, and the standard deviation of each sensor was 1. For classifiers which require fixed-size input vectors, feature vectors were created by resampling each instance into 20 timesteps of sensor values, and then by concatenating those 20 timesteps one after the other.

The implementations of MLPs, SVMs, and BNs that we used in these experiments come from the WEKA library [63]. In order to reproduce the results we report in this section on those classifiers, one must thus refer to the default parameters of their WEKA implementations. For some of the main parameters that we controlled in our experiments, we report the values we used at the bottom of each table.

3.4.1 Activity recognition performances

In this section, we compare the activity recognition performances of our place-based approach with a global approach, using different classical classifier types and decision fusion methods. Our goal is to study the validity of Hypotheses 3.1, 3.2, and 3.3, and if they are valid, the conditions that are required for this validity.

We have no guarantee that all activity classes will be equally as frequent in smart homes; in fact, it is highly probable that some classes are more frequent than others. Using simple non-weighted accuracy to compare activity recognition approaches can thus lead to spurious comparisons, in cases where a few activity classes are much more frequent than others in the datasets we use: in those cases, a classifier can have great accuracy by performing well on those frequent classes, even if it performs terribly bad on other classes. Such a situation is not desirable to provide context-aware services in smart homes, since this would mean that service quality would highly depend on the activity of the occupant, instead of being stable regardless of the activity.

Therefore, we use a weighted F_1 score (which we will simply refer to as F_1 score in the rest of the thesis) to compare performances, instead of simple accuracy. This performance measure gives equal weight to classes, regardless of the number of instances they contain. Given a set of activity instances $\mathcal{X}_a$ with

true label a for each activity class $a \in \mathcal{A}$, this F_1 score is computed as:

$$\mathrm{F}_1 = \sum_{a \in \mathcal{A}} \frac{2}{|\mathcal{X}_a|} \cdot \frac{\mathrm{Precision}(a) \cdot \mathrm{Recall}(a)}{\mathrm{Precision}(a) + \mathrm{Recall}(a)}, \tag{3.25}$$

where $\mathrm{Precision}(a)$ is the ratio between the number of instances of a correctly classified as a and the number of all instances classified as a, and where $\mathrm{Recall}(a)$ is the ratio between the number of instances of a correctly classified as a and the number of all instances of class a.

In the following subsections, the best results are typeset in bold in tables.

3.4.1.1 Performances on the Opportunity dataset

For reference, some of the results reported in this section were previously published in [36].

The Opportunity dataset, which was presented in Section 2.3.1.1, is the first dataset of activities in the home that we use to evaluate the recognition accuracy of our proposed place-based approach. This dataset presents the benefit of exposing the data of both wearable sensors as well as ambient sensors, which will allow to evaluate the behaviour of our approach in situations where both such sensor categories are recorded by the smart home system.

As mentioned in Section 2.3.1.1, the Opportunity dataset offers 5 different labels at each recorded timestep. Although activity labels are available, we argue that too few instances of those activity classes are available in the Opportunity dataset for a proper experimental evaluation. In the following experiments on the Opportunity dataset, we will thus study the accuracy of our place-based approach with respect to the *action* labels, and not the activity labels.

There are 17 classes of such actions labelled in the Opportunity dataset. In addition, an 18^{th} class, labelled "None", corresponds to instances during which no action is performed. Locations of action classes and of sensors, which are required to apply our place-based approach, are not explicitly given in this dataset. Nevertheless, we identified 3 distinct places in the experimental environment in which the Opportunity dataset was recorded: the *Table*, the *Kitchen*, and the *Exits* (see Figure 3.10).

Action classes and sensors [1] are thus distributed among those 3 places in the following way:

— Table: contains all 12 sensors attached to the objects placed on the table, as well as the 19 sensors worn by the occupant. The following action classes can occur in this place: "Clean table", "Drink from cup", "None".

— Kitchen: contains the 18 sensors attached to the fridge, 3 drawers, dishwasher, and the light switch, as well as the 19 sensors worn by the occupant. The following action classes can occur in this place: "Open fridge", "Close

1. Location tags data as well as quaternions data were not used due to reported noisiness in the documentation of the dataset and related papers.

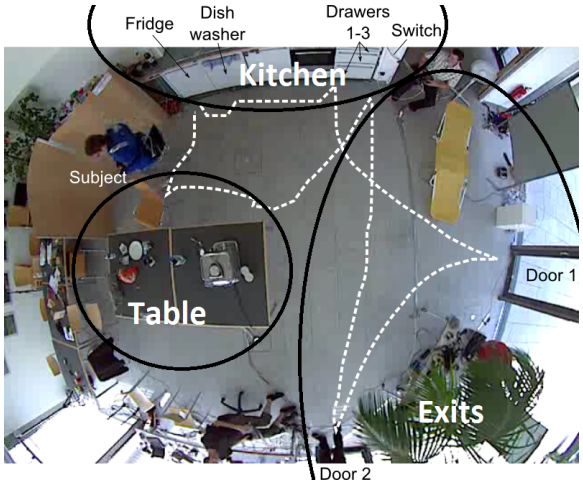

Figure 3.10 – The 3 places we identified in the environment where the Opportunity dataset was recorded: the *Table*, the *Kitchen*, and the *Exits*.

fridge", "Open dishwasher", "Close dishwasher", "Open drawer 1", "Close drawer 1", "Open drawer 2", "Close drawer 2", "Open drawer 3", "Close drawer 3", "Toggle switch", "None".

— Exits: contains the 2 sensors attached to the 2 exit doors, the sensor placed on the lazy chair next to one of the doors, as well as the 19 sensors worn by the occupant. The following action classes can occur in this place: "Open door 1", "Close door 1", "Open door 2", "Close door 2", "None".

We evaluate our approach using a 10-fold random cross-validation. Each action class of each fold contains 72 training instances, 22 test instances, and 18 validation instances. Those 3 sets of instances are used as is typically done in machine learning: the training set is used to learn the model. The test set is used to evaluate the performances of our approach on data that was never seen during training. The validation set is used to select the best parametrization of each classifier or decision fusion method: the learned models are first evaluated on the validation set; the parameters which yield the best performances on this validation set are used for the evaluation of the model on the test set. This allows to avoid a bias in the evaluation phase, by not optimizing the parameters of each model for the instances that we compare our models on. Instances are selected randomly among the 4 occupants recorded in the dataset.

Performance of classifiers in each place We report in Table 3.1 the F_1 scores of 4 classical classifiers (MLP, SVM, BN, and DTW) on each of the 3 different places (Table, Kitchen and Exits) in the Opportunity dataset. This first set of results allow us to observe a disparity in performances depending on classifier types. For example, we see that the DTW is significantly worse than the other 3 classifiers in the place Kitchen ($84.58\% \pm 1.38\%$ compared to $91,79\% \pm 1.27\%$

	Classifier			
Place	MLP	SVM	BN	DTW
Table	**98.97% ± 0.48%**[1]	98.77% ± 0.54%[2]	98.70% ± 0.48%	98.50% ± 0.44%
Kitchen	**94.06% ± 1.58%**[1]	93.78% ± 1.32%[2]	91.79% ± 1.27%	84.58% ± 1.38%
Exits	99.15% ± 0.39%[1]	**99.24% ± 0.34%**[2]	98.34% ± 0.62%	98.25% ± 0.81%

Parameters:

[1] 80 hidden neurons, 100 epochs, 0.2 learning rate, 0.1 momentum.

[2] $C = 1000$, $\gamma = 0.01$.

Table 3.1 – F_1 scores of classifiers for each place in the Opportunity dataset.

for the BN). Those results also show that classification performances can highly vary between places. Here, recognizing actions in the Kitchen seems significantly more difficult, for all classifiers, than recognizing actions in the 2 other places (e.g. 94.06% ± 1.58% compared to 98.97% ± 0.48% and 99.15% ± 0.39% for the MLP).

Such a gap could be in part explained by the number of input sensors of a place; in this case however, the Kitchen has a similar number of sensors as the other 2 places. It could be in part explained by the number of different classes to model; in this case, 12 action classes can occur in the Kitchen, which is indeed more than the 3 classes in place Table and 5 classes in place Exits. Finally, it could be in part explained by the complexity of modelling the classes from the sensors available; in this case, there are multiple action classes that will lead to similar sensor data: for example, wearable sensors will most likely produce similar data for both actions "Open drawer 1" and "Open drawer 2".

Comparison between the place-based and global approaches We report in Table 3.2 the F_1 scores of the place-based approach, when all places are modelled by the same classifier type, and of the global approach where one of the 4 previously mentioned classifier type is used to globally model actions in the Opportunity dataset. These results show that the place-based approach is on average better performing than the global approach. However, the gap between the global and the place-based approach is not statistically significant, as the standard F_1 score deviation of every single classifier type is big enough that the intervals of F_1 scores partially cover one another (e.g. 92.52% ± 1.25% compared to 90.21% ± 1.62% for the MLP). For the BN, we even observe that the average F_1 score of the place-based approach is lower than in the global approach (89.14% ± 1.27% compared to 90.61% ± 1.37%).

On the other hand, we observe that the standard deviation, for each classifier type, is smaller in the place-based approach than in the global approach. This is most probably because of the decision fusion step, which tends to average out the results.

	Classifier			
Approach	MLP	SVM	BN	DTW
Global	$90.21\% \pm 1.62\%$[1]	$90.05\% \pm 1.64\%$[2]	$\mathbf{90.61\% \pm 1.37\%}$	$75.03\% \pm 2.53\%$
Place-based	$\mathbf{92.52\% \pm 1.25\%}$[3]	$\mathbf{91.78\% \pm 1.37\%}$[4]	$89.14\% \pm 1.27\%$[5]	$\mathbf{83.55\% \pm 1.44\%}$[6]

Parameters for the global approach:

[1] 500 hidden neurons, 200 epochs, 0.2 learning rate, 0.1 momentum.

[2] $C = 100$, $\gamma = 0.0005$.

Decision fusion for the place-based approach (the parametrization of each place's classifier is reported in Table 3.1):

[3] SVM stacking with $C = 100$, $\gamma = 0.01$.

[4] MLP stacking with 100 hidden neurons, 100 epochs, 0.2 learning rate, 0.1 momentum.

[5] MLP stacking with 20 hidden neurons, 100 epochs, 0.2 learning rate, 0.1 momentum.

[6] SVM stacking with $C = 1$, $\gamma = 0.1$.

Table 3.2 – F_1 scores of classifiers using the global approach or the place-based approach on the Opportunity dataset.

	Classifier		
	MLP	SVM	BN
Approach	Fusion		
Global	$91.62\% \pm 1.59\%$[1]		
Place-based	$\mathbf{92.70\% \pm 1.26\%}$[1]		

Classifiers' parameters: see Table 3.1 and Table 3.2.
Decision fusion parameters:

[1] SVM stacking with $C = 1$, $\gamma = 0.1$.

Table 3.3 – F_1 scores of decision fusion on three different classifier types for both the global approach and the place-based approach on the Opportunity dataset.

Multi-classifier fusion We report in Table 3.3 the F_1 scores of the place-based approach, when 3 different classifiers (MLP, SVM and BN) are fused for each place, and of the global approach, when those 3 same classifier types are fused. We did not include the DTW in this experiment, as its previous results were significantly worse than the other 3 classifier types. We observe that, as with the results reported in Table 3.2, the place-based approach is on average better than the global approach but with no real statistical significance ($92.70\% \pm 1.26\%$ compared to $91.62\% \pm 1.59\%$). Moreover, we see that using 3 different classifiers, instead of just one, leads to marginal performance improvements. For example, the place-based approach with one MLP per place obtained the best previous performance with an F_1 score of $92.52\% \pm 1.25\%$ (as reported in Table 3.2), whereas the place-based approach with 3 classifiers per place obtained a score of $92.70\% \pm 1.26\%$.

Confusion between classes We report on Figure 3.11 the confusion matrix of one test fold of the multi-classifier place-based approach, which obtained the overall best results (92.70% $\pm$ 1.26% F_1 score). We can observe two distinct phenomena: first, we see that action classes that are intuitively close to each other, such as "Open drawer 1", "Close drawer 1", "Open drawer 2", etc., lead to the most confusion. Indeed, as discussed previously with the results reported in Table 3.1, such actions will most likely generate similar wearable sensor data. Considering that, in all 3 places, there is a majority of wearable sensors (12 ambient and 19 wearable in Table, 18 ambient and 19 wearable in Kitchen, 3 ambient and 19 wearable in Exits), it is expected to see that even the best approach has trouble differentiating between each of those action classes.

The second phenomenon we can observe from this confusion matrix is that the other main source of confusions seems to be the class "None". As anticipated in Hypothesis 3.3, this class is difficult to model for a place-based approach because it correspondents to situations where nothing is happening in that place in particular, which can be many different classes of actions in other places and thus many different patterns of sensor data. Indeed, in a situation where the occupant performs an action in one place, other places should recognize class "None", despite potentially observing sensor data that occurs in other actions. This problem is exacerbated on the Opportunity dataset considering the number of wearable sensors, which are thus the main source of information of each place during training.

Discussion This first set of experiments we reported on the Opportunity dataset sheds some light on the performances of our proposed place-based approach. First, we observed that some places will be harder to perform well on than others, for a variety of reasons such as the number of data sources, the number of target classes, or the similarity between classes. Through our place-based approach, it is possible to work on specifically enhancing the performances on one place, while ignoring other places where the place-based approach performs well; this is not possible with a global approach in which attempts at improving performance on one place will have unpredictable effects on the performance on other places. Our proposed place-based approach is thus probably well-adapted in situations where some of the places in the home are harder to model than others.

The results we obtained showed no statistically significant difference in performance between our place-based approach an the global approach, albeit the place-based approach was on average slightly better. However, the place-based approach seems to be more stable than a global approach, as the standard deviations reported are quite smaller. Using multiple classifiers per place seem to not have much impact on the performances of the place-based approach.

We think that the high number of wearable sensors in the Opportunity dataset can be a major reason as to why using a place-based approach did not improve performances compared to a global approach. Indeed, since our place-

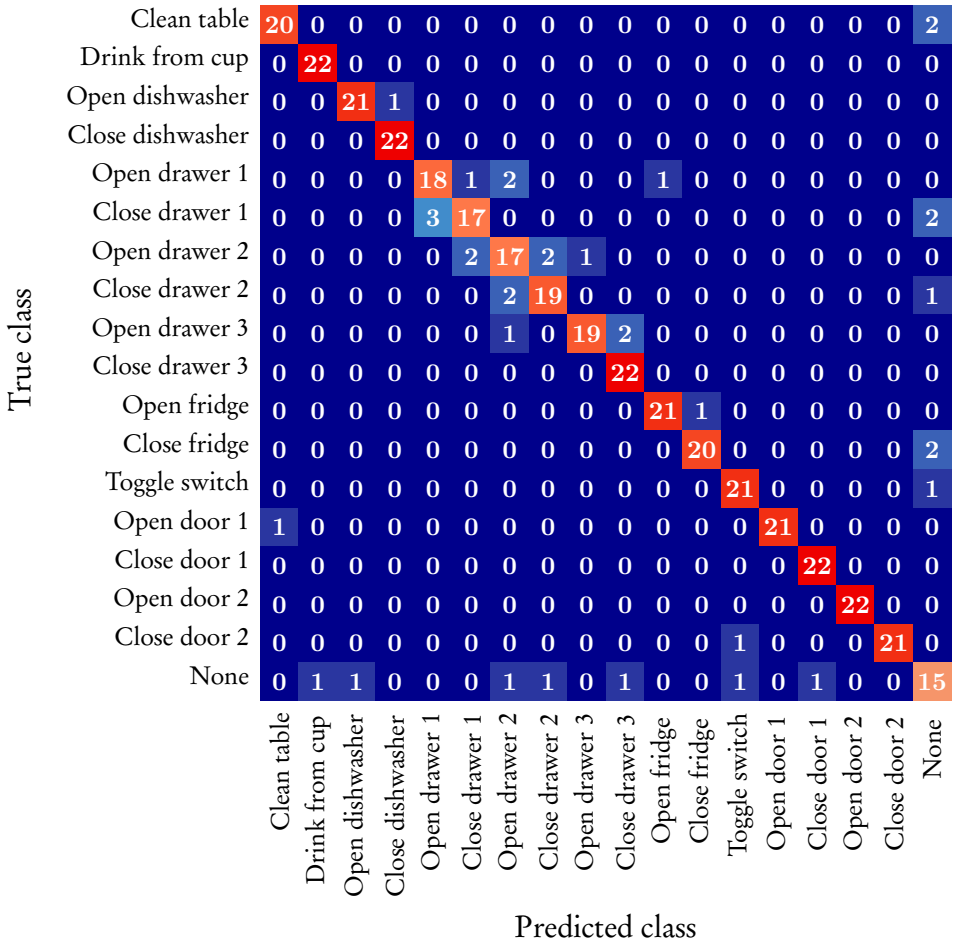

Figure 3.11 – Confusion matrix of one fold of test of the place-based approach reported in Table 3.3 on the Opportunity dataset.

based approach does not presuppose knowledge of the location of the occupant, each place has to use the wearable sensors as data sources. In this dataset, each of the 3 places has a majority of wearable sensors (12 ambient and 19 wearable in Table, 18 ambient and 19 wearable in Kitchen, 3 ambient and 19 wearable in Exits). Therefore, the main source of data for a place are sensors which provide information both when the occupant is in that place but also when they are not: this can easily lead to confusions, especially with class "None", as was shown on Figure 3.11 where the place-based approach would often either mistaken an action as "None" or mistaken "None" for an action. The specific problems related to class "None" may be addressed using instance rejection as in [18], or by modelling the dissimilarity between "None" and other classes (not addressed in this thesis).

Therefore, we propose to evaluate the recognition performances of our place-based approach on Orange4Home, which, as presented in Section 2.3.2, is a

dataset that does not contain any wearable sensors.

3.4.1.2 Performances on the Orange4Home dataset

For reference, some of the results reported in this section were previously published in [37].

The Orange4Home dataset was recorded during 4 consecutive weeks, in much the same way data from a real smart home system would be collected. As such, we chose to evaluate the performances of our place-based approach not through a cross-validation protocol, but rather using a more realistic training protocol that could be used in a real smart home system: the first few days or weeks of data are used to train the model, such that it can be operational as quickly as possible; the performances of this model are then judged on future data, in the temporal sense (and not, as in a cross-validation protocol, on future data in the sense that it has not been seen yet, but disregarding when it was collected compared to training data). As such, we decided to use the first 2 weeks of data to train models on the Orange4Home dataset, the 3^{rd} week of data as a validation set, and the 4^{th} week of data as a test set.

Comparison between the place-based and global approaches We report in Table 3.4 the F_1 scores of the place-based approach and the global approach when only one classifier type (either the MLP or the SVM) is used (first two columns of the table). We observe this time a larger gap between the place-based approach and the global approach: the place-based approach obtained an F_1 score of 93.05% using MLPs in each place, whereas the global approach with one large MLP obtained a score of 77.85%. The same observation can be done for the SVM, albeit with a smaller margin (92.08% compared to 89.61%).

These results corroborate our suggestion that wearable sensors negate the benefits of our place-based approach. On the Orange4Home dataset where such sensors are not present, our place-based approach seems significantly more accurate. Moreover, the Orange4Home dataset contains significantly more sensors, places, and activity classes (236, 8, and 27 respectively) than the Opportunity dataset (52, 3, and 18 respectively). Therefore, it will be harder for a global approach to properly model the relationship between sensors and activity classes on the Orange4Home dataset, whereas the place-based approach is less insensitive to the number of sensors and activity classes if the number of places is high enough, as described in Section 3.3.6.

Multi-classifier fusion We report in the last column of Table 3.4 the F_1 socres of the place-based approach and of the global approach when both the MLP and SVM are used simultaneously. Much like the results reported in Section 3.4.1.1, we observe that using multiple classifiers per place marginally improves the performances of the place-based approach (93.29%) compared to the individually best classifier type (93.05% for the MLP).

	Classifier		
			MLP SVM
Approach	MLP	SVM	Fusion
Global	77.85%	89.61%	89.61%[1]
Place-based	**93.05%**[2]	**92.08%**[3]	**93.29%**[2]

Decision fusion for the global approach:

[1] Possibility theory fusion

Decision fusion for the place-based approach:

[2] DST fusion

[3] Majority vote fusion

Table 3.4 – F_1 scores of classifiers using the global approach or the place-based approach on the Orange4Home dataset.

Confusion between classes We report in Table 3.5 the list of confusions made by the multi-classifier place-based approach on the Orange4Home dataset during the test phase, for which it obtained an F_1 score of 93.29%. We can observe that in the majority of cases (5 times out of 7), the right place is recognized even though the activity class itself is not. If misclassification errors were distributed uniformly at random among places, we would not observe the same bias: for the first instance ("Cleaning" in the Bathroom), 3 other classes can occur in the same place ("Showering", "Using the sink", and "Cleaning"), and 26 classes other than "Cleaning" in the Bathroom can occur globally in the home. As such, this instance would have a $3/26$ probability of being misclassified as an activity class that occurs in the same place. The same reasoning for the other 6 misclassified instances leads to a probable number of instances misclassified as an activity class that occurs in the same place of $(6 \times 3 + 2)/26 \approx 0.77$, which is significantly less than the 5 cases we observe here. This indubitably shows that our place-based approach is biased towards same-place misclassifications: if the approach misclassifies an activity instance, it will likely classify it as an other activity that occurs in the same place. This bias can actually be beneficial in smart home systems, as we will argue in Chapter 5.

3.4.1.3 Conclusions on recognition performances

We first showed through these experiments that a place-based approach is on average better performing than a global approach on the 2 datasets we used, as hypothesized in Hypothesis 3.1. Although this difference is relatively insignificant on a dataset where wearable sensors are prominent, the gap is much larger on a dataset containing exclusively ambient sensors, which are the desired types of sensor installations in general public smart home systems as argued in Section 2.2.

We then showed that multi-classifier fusion leads to better performances

Ground truth		Prediction	
Place	Class	Place	Class
Bathroom	Cleaning	Bedroom	Dressing
Bathroom	Using the sink	Bathroom	Using the toilet
Bathroom	Using the sink	Bathroom	Using the toilet
Entrance	Leaving	Bedroom	Napping
Kitchen	Preparing	Kitchen	Cleaning
Kitchen	Preparing	Kitchen	Cleaning
Kitchen	Preparing	Kitchen	Cleaning

Table 3.5 – Confusions made by the multi-classifier place-based approach on the Orange4Home dataset in the test phase.

for both place-based and global approaches, as anticipated in Hypothesis 3.2. However, this improvement is very marginal and it can be argued that it is not worth the added complexity. Nevertheless, if computing power is not limited, multi-classifier fusion is a free improvement, as it requires no additional data sources.

Finally, we showed that, when activity "None" is part of the set of activity classes to recognize, this class generates a lot of confusions and is hard to model. In particular, when many wearable sensors are used as data sources, a place-based approach will be even more confused. Indeed, when the occupant performs an activity in one place, other places will observe all the changes reported by the wearable sensors, and thus be confused into thinking that something is happening in the place they are monitoring too. Hypothesis 3.3 is thus verified, and exacerbated in situations with wearable sensors.

3.4.2 Computing times

For reference, some of the results reported in this section were previously published in [36].

Beyond recognition performances, the other benefits of the place-based approach we anticipated are related to computing times. We conjectured in Hypotheses 3.4 and 3.5 that both the training phase and activity recognition at run time will be faster with the place-based approach compared to the global approach. To verify this hypothesis, we will study the computing times of these phases that we recorded during the experiments on recognition performances. All of the computing times we report in this section were recorded on a 4-cores Intel® Core™ i7 2.8 GHz processor with 16 GB of RAM.

Computing times on the Opportunity dataset We report in Table 3.6 the training and test times of 3 different types of classifiers (MLP, SVM, BN) for

each of the 3 places and for the global approach on the Opportunity dataset. The computing times of the decision fusion step (regardless of the decision fusion method used) are negligibly small compared to the other computing times; they are thus not reported in the table and ignored in our analysis. We will ignore the DTW in the following discussions, even though it requires no training time, as it is much too slow compared to the other 3 classifier types in the test phase.

Assuming that our place-based approach is executed on a multi-core computing architecture (either on a multi-core processor, or on multiple computing devices in a network), we can parallelize the training phase or the recognition phase at run time of our place-based approach: all places are computed simultaneously (which therefore requires as many cores as there are places); the computing time of the place-based approach is thus the computing time of the slowest place (assuming decision fusion and parallelization overheads take a negligible amount of time). In our experiments, the computing times of our place-based approach are thus always significantly shorter, both in the training and test phase, than a global approach. For example, training a place-based approach using MLPs will on average take 947.65 seconds (because place Table is the slowest one), whereas training a global approach that uses a MLP on average take 11250.06 seconds, which is more than 10 times longer. Similarly, recognizing at run time all test activity instances will take on average 12.49 seconds for the place-based approach using SVMs (because place Kitchen is the slowest one), when it will take a global approach that uses a SVM 29.47 seconds.

In the multi-classifier scenario, the same behaviour can be observed. In the global approach, the 3 classifiers can be parallelized; we thus need on average as much time as the slowest of these classifiers (for example, 11250.06 seconds for training because of the MLP). As for the place-based approach, we can parallelize the places; we thus need on average as much time as the slowest place, which computing time is the sum of the computing times of the 3 classifiers (for example, $947.65 + 24.42 + 19.06 = 991.13$ seconds because place Table is the slowest). We see that the place-based approach is thus once again faster than a global approach in this experiment, even in the test phase: the global approach takes 29.47 seconds to complete (because the SVM is the slowest), while the place-based approach takes $9.84 + 12.49 + 6.71 = 29.04$ seconds (because place Kitchen is the slowest).

Since there are 396 instances of activities per fold, any of the approaches experimented here will be able to process an activity instance at run time in a time much shorter than the duration of an activity instance itself. For example, our multi-classifier place-based approach will take on average $29.04/396 = 0.073$ seconds to process an activity instance, which is sufficiently short to be unnoticeable by an occupant.

Computing times on the Orange4Home dataset We report in Table 3.7 the training and testing times of 2 different types of classifiers (MLP, SVM) for each of the 8 places and for the global approach on the Orange4Home dataset. If we

Classifier	Phase	Model			
		Table	Kitchen	Exits	Global
MLP	Training	947.65 ± 160.77	732.83 ± 60.04	561.71 ± 30.78	11250.06 ± 1593.57
	Test	12.64 ± 1.56	9.84 ± 1.22	8.49 ± 1.99	20.70 ± 1.19
SVM	Training	24.42 ± 0.23	19.11 ± 0.16	12.75 ± 0.23	35.37 ± 0.48
	Test	6.56 ± 0.06	12.49 ± 0.13	4.21 ± 0.03	29.47 ± 0.96
BN	Training	19.06 ± 0.34	13.87 ± 0.28	11.34 ± 0.25	26.49 ± 0.37
	Test	8.75 ± 0.06	6.71 ± 0.13	5.49 ± 0.07	11.73 ± 0.10
DTW	Training	0	0	0	0
	Test	4116.70 ± 262.37	3256.30 ± 199.21	2937.00 ± 174.20	5011.10 ± 335.78

Parameters: see Table 3.1 and Table 3.2.

Table 3.6 – Average computing times (in seconds) of classifiers during the training and test phases for each model, for an entire fold of cross-validation on the Opportunity dataset.

assume that we have as many computing cores as the number of places (here, 8), we arrive to the same conclusions on the Orange4Home dataset as we did on the Opportunity dataset: the training and test times of the place-based approach are significantly shorter than a global approach. For example, it takes 23.73 seconds to train the place-based approach using MLPs (because place Kitchen is the slowest), when it takes 319.34 seconds for a global MLP.

However, having as many computing cores as the number of places (for example 8 on the Orange4Home dataset) might not always be plausible. Nevertheless, we can still parallelize the place-based approach even if there are more places than available computing cores: for example, with 3 computing cores, we can spread the 8 places of the Orange4Home dataset among the 3 cores, such that 2 cores sequentially process 3 places each, and 1 core sequentially processes the remaining 2 places. The worst-case scenario is when the 3 places that are the longest to compute are computed by the same core. For example, the worst-case scenario when training MLPs in the place-based approach using only 3 cores is when the Kitchen, the Living room and the Bedroom are computed on the same core, which would take here $23.73 + 13.57 + 12.89 = 50.19$ seconds.

We report in Table 3.8 the worst-case computing times of the place-based approach on the Orange4Home dataset when we have 1 to 4 computing cores available. We see (in bold) that we can obtain shorter computing times than the global approach (as reported in Table 3.7) in all cases with only 4 computing cores, which is 2 times less than our first assumption of 8 cores. With only 3 cores, the place-based approach is faster than the global approach in all but the SVM training case, where it is only 1.34 seconds slower (29.39 seconds compared to 28.05 seconds).

The previous computing times occur in worst-case scenarios. In a real system, some simple criteria would allow to optimize the partition of places among computing cores, such that the maximum time required by any of the cores is minimal: for example, the more sensors are in a place, or the more activity classes

	Classifier			
	MLP		SVM	
Model	Training	Test	Training	Test
Entrance	7.00	2.10	7.42	2.02
Kitchen	23.73	3.05	10.87	2.92
Living room	13.57	2.66	9.56	2.56
Toilet	5.92	1.93	6.83	2.02
Staircase	5.37	1.89	6.50	1.80
Bathroom	11.53	2.39	8.24	2.43
Office	9.72	2.23	7.92	2.15
Bedroom	12.89	2.56	8.96	2.47
Global	319.34	9.81	28.05	10.20

Table 3.7 – Average computing times (in seconds) of classifiers during the training and test phases for each model on the entirety of the Orange4Home dataset.

can occur in a place, the more complex the model will be for this place. Therefore, we could share places based on those criteria such that places with many sensors and activity classes and places with few sensors and activity classes are computed on the same core, while places with an average number of sensors and activity classes are grouped together; this way, no computing core will be significantly slower than the rest. Using this method, it seems that we can divide by at least a factor of 2 the number of computing cores required to parallelize our place-based approach, compared to the number of places.

3.4.2.1 Conclusions on computing times

In this second set of experiments, we observed that, as anticipated in Hypotheses 3.4, the training times required by a place-based approach are significantly smaller than for a global approach, for any of the tested classifier types. In particular, for some classifier types like the MLP, where training times increase greatly with the number of inputs and output classes, this gap can be very large. In much the same way, the execution times at run time of a place-based approach are also smaller than for a global approach, as conjectured in Hypothesis 3.5.

However, for these hypotheses to be valid, parallelization is necessary, using as many computing cores as places in the home. Yet in practice we showed that we can divide the number of required cores by at least a factor of 2 in the worst-case, and thus by a bigger factor using simple heuristics for distributing places on computing cores based on the number of input sensors and output classes of each place. Parallelization capabilities is thus not a very limiting barrier. Approaches such as multi-agent systems and fog computing could be applied to orchestrate this place-based parallelization.

Number of cores	Classifier			
	MLP		SVM	
	Training	Test	Training	Test
1	**89.73**	18.81	66.30	18.37
2	**61.72**	10.66	37.63	10.38
3	**50.19**	**8.27**	29.39	**7.95**
4	**37.30**	**5.71**	**20.43**	**5.48**

Table 3.8 – Average computing times (in seconds) of the place-based approach on the Orange4Home dataset depending on the number of available computing cores, in worst-case scenarios where all slowest places are running on the same core. Times typeset in bold are those that are smaller than the corresponding times for a global approach.

3.5 Conclusions

We presented in this chapter the second contribution of our thesis: place-based activity recognition. We experimentally evaluated this approach on two different datasets and showed significant improvements, compared to global approaches, in activity recognition performance as well as computing times.

Improvements in activity recognition performance are essential in order to increase the usefulness of context-aware smart home systems to occupants. Incorrectly identifying situations can lead to inappropriate services, which is unacceptable for general public users. As such, our place-based approach is more well-adapted to such general public smart home systems.

Improvements on computing times are also valuable. Indeed, low computing complexity means that activity recognition can possibly run on relatively cheap hardware, including objects that are part of the sensor network themselves, instead of expensive home automation boxes or cloud-based solutions. This implies in particular that our place-based approach is applicable for local smart home solutions that do not process personal data in the cloud, which is generally desired for privacy reasons.

We argued that our place-based approach, because of its modularity, is well-adapted to the current ecosystem of smart home technologies and scientific improvements in AI. Indeed, this approach is agnostic to the classifiers used in each place or the sensors installed. Evolutions in the sensor installation or the routines of occupants can be integrated by the approach at the place level, instead of having to modify the entire model of the home. Moreover, this modularity also means that simultaneous activity recognition in different places is a natural perspective for the place-based approach, whereas it is unclear what a good strategy for this would be when starting from global approaches.

Recognizing activities using our place-based approach is the first necessary step to provide our example service of communication assistance to occupants, assuming that availability for communication highly depends on the activity of occupants. However, this service should not only be able to estimate the current availability of occupants, but also predict their future availabilities, in order to give recommendations to correspondents. In Chapter 4, we thus study the problem of activity prediction, from a series of previous activity instances recognized for example by the place-based approach. We present new context-based prediction models that take the specificities of smart home environments into account. We evaluate those contributions on state-of-the-art smart home datasets, as well as on the Orange4Home dataset.

CHAPTER 4

Predicting activities using context

Future context information is essential to allow a smart home to anticipate the needs and behaviour of its occupants. Such anticipation is necessary to provide a number of context-aware services, such as a communication assistant that advises outsiders about the future availability for communication of occupants of the home. Therefore, smart home systems must be able to predict future context situations, and thus, must be able to predict future activities that its occupants will perform. Following preliminary assumptions and a more precise definition of the activity prediction problem, presented in Section 4.1, and a survey of activity prediction strategies for smart home published in the state of the art, presented in Section 4.2, we present in Section 4.3 the third major contribution of our thesis, the PSINES (short for Past SItuations to predict the NExt Situation) Dynamic Bayesian Network (DBN), and intermediary contributions, which seeks to improve the prediction accuracy of a state-of-the-art algorithm using context information and smart home specificities. We experimentally evaluate this new approach in Section 4.4.

4.1 Problem statement and preliminary assumptions

Before presenting our contributions on the problem of activity recognition, we first need to define what we mean precisely by activity prediction. This is the object of Section 4.1.1. We also state, in Section 4.1.2 the assumptions we make about to simplify this problem.

Sequence prediction Occurrence prediction

Figure 4.1 – Comparison between predicting the next activity label at timestep $i+1$ in a sequence, and predicting the next occurrence time t_{i+1} of a particular class A_n.

4.1.1 Activity prediction

In this chapter, we study the problem of activity prediction. However, we find, in the literature, many different uses of the term "activity prediction", that refer to seemingly different problems. In cause is the word "prediction" which is quite often overloaded, especially in the research field of machine learning. Merriam-Webster provides the following definition of the verb "to predict":

> **Definition: Predict (Merriam-Webster [3])**
> *To declare or indicate in advance; especially: foretell on the basis of observation, experience, or scientific reason.*

This definition clearly includes a notion of precedence, that is, that the prediction occurs in *advance*, or in other words that the object of the prediction will occur after the prediction itself.

This definition clearly does not fit the use of "activity prediction" to designate a number of studied problems which do not include this notion of precedence. One such problem is that of activity recognition, which we studied in Chapter 3. In papers related to this problem, "predicting" is often used to mean "recognizing" or "classifying". This is not the problem we study in this chapter.

Another such problem is that of recognizing activities as early as possible using partial data. Multiple papers refer to this problem as "activity prediction", such as [130, 65, 92]. This problem, however, is more akin to activity recognition using the entire instance data, since the goal is to affect the activity label that best fit the current recorded data, even if future data might still represent the same instance. This is not the problem we study in this chapter.

We do find, in the literature of smart homes, 2 distinct problems that fit the definition of activity prediction. The first problem consists in predicting, for each possible activity class, the time of occurrence in the future of the next instance of that class. As far as we know, and as the authors claim in the paper, this problem was first introduced by Minor et al. in [102] (in which they call this problem "activity prediction"). Lago et al. also study this problem in [83]. Let us call this problem "occurrence prediction" for the remainder of this section.

The second problem consists in predicting the next future activity instance

that will occur after the current activity. This problem is studied in multiple papers such as [164, 72, 62, 111]. Let us call this problem "sequence prediction" for the remainder of this section.

Both problems seem, at first, to answer different questions: occurrence prediction allows to answer questions such as "when will the occupant cook next?", whereas sequence prediction allows to answer questions such as "what activity will the occupant do after he's done cooking?". Yet, we can show that sequence prediction can be solved using an occurrence predictor, while the opposite is not directly possible. In occurrence prediction, the algorithm predicts, for each activity class, the next time at which it will occur. Given this set of occurrence time for each activity class, we can derive the next activity that the occupant will perform (the sequence prediction problem) by observing which activity class will occur the soonest (assuming a closed-world scenario where the complete list of possible activity classes is known). Using a sequence predictor, we can recursively predict a sequence of future activities long enough such that it contains each activity class at least once. However, we would still not know the times at which these activities will occur, and would thus require an additional algorithm to solve the occurrence prediction problem.

As such, solving the occurrence prediction problem is more desirable than only solving the sequence prediction problem. On the other hand, the generality of the occurrence prediction problem can be problematic: sequence prediction, being a simpler problem, might be tractable using available smart home data, while occurrence prediction might not be. In particular, predicting time values intuitively seems more difficult than predicting activity labels. This difficulty might be one of the reasons why there are many more works on sequence prediction in the literature of smart homes compared to occurrence prediction.

For these reasons, we will study, in this chapter, the problem of sequence prediction, which we will simply call *activity prediction* in the rest of the thesis.

4.1.2 Assumptions

In the following section, we explicitly state some of the hypotheses we make about the activity prediction problem (as defined in Section 4.1.1). These assumptions were either already made for the activity recognition problem, or follow from our study of activity recognition. Justifications for some of these assumptions can thus be found in Chapter 3.

4.1.2.1 Single-occupant situations

We assume that only one person occupies the home at all times (although that person can change). This assumption was already made for activity recognition in Section 3.1.1.

4.1.2.2 Information on identity

We assume that the context dimension of identity is given before the prediction step, or equivalently, that the identity of the occupant is not considered important for activity prediction. This assumption was already made for activity recognition in Section 3.1.2.

4.1.2.3 Existence of an activity recognition model

We assume that we have at our disposal an accurate activity recognition model, which can identify activity instances given sensor data. This assumption allows us to potentially predict activities using previously recognized activities, instead of having only raw sensor data at our disposal. This assumption can be met using for example the place-based activity recognition approach studied in Chapter 3 (this approach also provides information about place, which can then also be used for activity prediction).

4.1.2.4 Sequentiality of activities

We assume that activity instances necessarily follow one another sequentially. In other words, we assume that two activities cannot occur simultaneously, i.e. the occupant never performs two activities in parallel. This assumption was already made for activity recognition in Section 3.1.4.

4.2 State-of-the-art approaches

In the following section, we present in more details a number of contributions on the problem of activity prediction. We mostly focus on works applied to smart environments. We base our survey on the definition of activity prediction we chose in Section 4.1.1. As such, we will review contributions where the goal is to predict the class of the future activity instance based on current and past situations and sensor data.

Similarly to the problem of activity recognition, machine learning techniques are often used for activity prediction. Their ability to automatically learn the right model from labelled data constitutes an obvious advantage. Moreover, such labelled datasets are also required for activity recognition based on machine learning techniques. Therefore, a machine learning algorithm for prediction can benefit from the same labelled training dataset that was constructed for the purpose of the preliminary activity recognition step.

In addition, we find a large number of contributions related to the manipulation of sequences of symbols: sequence matching, compression algorithms, itemsets mining, etc. Such techniques, originally applied to classical problems on data compression, database exploration, data mining, etc., can be applied to activity prediction in smart homes fairly straightforwardly.

In Section 4.2.1, we discuss works based on sequence mining techniques. In Section 4.2.2, we present contributions that combine both mining and machine learning algorithms. In Section 4.2.3, we review papers that propose machine learning approaches for activity prediction. Finally, in Section 4.2.4, we discuss the benefits and drawbacks of each approach, in anticipation for our own contributions on activity prediction, in Section 4.3.

4.2.1 Sequence mining

A major category of activity prediction algorithms studied the literature of smart home research are related to the manipulation of sequences of symbols. Sequence matching, compression, itemsets mining, pattern mining, association rule mining, etc., are all example of general categories of algorithms applied to the problem of activity prediction in smart homes. In fact, in a survey of activity prediction in homes by Wu et al. in 2017 [161], we exclusively find algorithms from these categories.

This predominance is understandable due to the long history of research on the extraction of patterns in sequences of symbols, which reach all the way back to Shannon's work on the entropy of information [137]. A large number of well-established algorithms for compression, itemsets mining, pattern mining, etc., that were designed for other applications, are thus applied to the problem of activity recognition in homes. Classical examples of such algorithms include *Apriori* [6], *Minepi* and *Winepi* [95], FP-Growth [64], Eclat [168], etc. In Apriori, the discovery of frequent itemsets relies on the property that the subset of a frequent itemset is necessarily frequent. As such, one can first establish the list of frequent (according to some support value) itemsets of 1 element, and then recursively generate new candidate itemsets, with one more element, from the previous list of frequent itemsets, avoiding the generation of necessarily infrequent itemsets. Many related algorithms rely on similar generation techniques. For example, in order to discover episodes in temporal data, the Winepi algorithm consists in applying Apriori following a preliminary step that turns a temporal sequence of data into a discrete transaction using a fixed-size sliding window.

We thus find a number of different papers that employ such techniques for activity prediction in smart homes [92, 96, 62, 39, 110, 27, 147]. We discuss in more details 2 papers that present such approaches in Section 4.2.1.1 and Section 4.2.1.2.

4.2.1.1 Active LeZi

Gopalratnam and Cook propose in [59] a sequence prediction algorithm that relies on data compression techniques, called Active LeZi. This predictor is based on the LZ78 text compression algorithm, which incrementally parses sequences of symbols to construct a dictionary of symbol phrases. Drawbacks of LZ78 include slow convergence, which was already tackled by Bhattacharya and Das

in [20] with the LeZi update technique. Active LeZi is an improvement on both LZ78 and LeZi update to address convergence slowness, by using sliding windows. The authors prove a number of theoretical results on the convergence rate and complexity of Active LeZi, compared to previous solutions.

The main application domain Gopalratnam and Cook apply Active LeZi to is smart home. In particular, they consider their approach suitable for predicting activities of occupants, which is useful for services such as home automation. In their experiments, they study the behaviour of Active LeZi, not directly on human activity prediction, but rather on interaction prediction. The goal is to predict, from a sequence of previous events, the next change observed by a device (e.g. a door sensor detects that its door is now opened). They show that Active LeZi can obtain reasonable prediction performance, especially when the top five predictions made by the algorithm are considered.

This first work shows that compression-based techniques can be applicable to event prediction in smart homes, and possible directly to activity prediction (although it was not tested in experiments). The large body of work on compression techniques and identification of symbol sequences in general can thus be exploited for prediction problems in smart homes.

4.2.1.2 SPEED

In [9], Alam et al. present a sequence prediction algorithm called SPEED. Previous compression techniques such as Active LeZi, presented in Section 4.2.1.1, are based on general sequences of data and do not take into accounts the properties of smart home data sources and specificities in human behaviour. SPEED, on the other hand, is designed specifically for the problem of activity prediction in smart homes, and thus takes into consideration certain properties. For example, as this work limits itself to "On/Off" sensors, we can identify episodes of data collected between an "On" event and a "Off" event of one sensor, that correspond to home interactions.

SPEED constructs a decision tree which contains the identified episodes in a sliding window over the data sequence. This decision tree corresponds to a finite-order Markov model. This decision tree can be used to infer the probability of occurrence of a specific symbol after having observed a specific window of data. Alam et al. present a number of theoretical results on the time and space complexities of SPEED. In particular, they show that it converges faster than Active LeZi and leads to more accurate prediction, while requiring more data storage for the decision tree.

Efforts in improving convergence rates of sequence prediction algorithms allow the application of such techniques to smart home datasets. Indeed, we have seen in previous chapters that acquiring large amounts of labelled data in smart homes is not realistic. As such, algorithms which require less data to converge are more easily applicable in such situations. Moreover, such algorithms that comprehensively extract all patterns from data sequences can be prohibitively

slow on long sequences, which may limit their usefulness with regards to smart home services that require fast activity prediction times.

4.2.2 Machine learning prediction with preliminary sequence mining

In the previous section, we discussed works that use sequence matching, compression algorithms, and itemset mining for activity prediction. These papers benefit from the extensive body of work on symbol manipulation and data mining, and the theoretical grounds for such approaches.

These techniques are generally agnostic to the actual source of data on which they are applied. On one hand, this means that they can be applied on any sequence-based data regardless of the actual problem tackled. In particular, this means that they can be applied to smart home activity prediction, as we have seen in the previous section. On the other hand, this also means that these approaches cannot be straightforwardly modified to integrate *a priori* knowledge about home environments.

As such, some authors have proposed to use a combination of sequence matching techniques in conjunction with machine learning algorithms, which can adapt to the specificities of data sources. In this case, sequence matching is used to extract well-supported patterns in the data to construct features, which are then learned on by the machine learning algorithm to predict activities. We discuss in more details 2 such papers in Section 4.2.2.1 and Section 4.2.2.2.

4.2.2.1 Discovering behaviour patterns for activity prediction

Fatima et al. propose in [51] a two-step module for activity prediction in smart homes. In the first step, sequence pattern mining is applied in order to discover temporal patterns of activity sequences. A support threshold allows to prune sequences of activities that are too infrequent in the dataset. According to the authors, this step allows to find significant behaviour patterns that occur frequently on different days, and that thus may be used to model the routine of occupants.

In the second step, a Conditional Random Field (CRF) is employed to model activity sequences for activity prediction. CRFs are generative probabilistic graphical models that allow to capture directed dependencies between variables, such as activities. Sufficiently supported sequences of activities found in the first step are used to train the CRF. In particular, Fatima et al. propose to only use supported sequences of 8 to 10 consecutive activities to predict future activities, and discard shorter supported sequences. They show in their experiments that their two-step method leads to better prediction accuracy compared to a HMM.

In this paper, it is argued that, in order to predict future activities, it is necessary to observe long sequences of past activities. Using such sequences of 8 to 10 activities seem to allow graphical models such as CRFs to model the

problem of activity prediction. In short, this work suggests that a future activity is determined not by its immediately preceding activity, but rather by a complete sequence of preceding activities, i.e. activities might not follow the first-order Markov property.

4.2.2.2 Itemset mining and temporal clustering for prediction

In [163], Yassine et al. propose a three-step process for activity prediction in smart homes. First, an itemset mining strategy is used to identify frequently supported patterns that relate activities to appliance usage in collected data. The extraction algorithm they use is based on FP-Growth and is presented in more details in [138].

While the first step focuses on finding appliance-to-appliance and appliance-to-activity relationships, the second step is used instead to find appliance-to-time relationships. In particular, the goal is to discover, in recorded data, the usage time of appliances with respect to various temporal elements: hour of day, time of day, weekday, week of the year, month of the year. An incremental k-means strategy is used to cluster appliances with similar temporal usages together.

Finally, the third step aims at predicting activities from these appliance-to-appliance frequent patterns and appliance-to-time relationships. In order to do so, they propose to use a BN in which these appliance and temporal associations are modelled in the BN's structure. The BN can predict the most probable future appliance usages, and thus, future activities based on appliance-to-activity relationships identified in the first step.

Contrary to the works of Fatima et al. presented in Section 4.2.2.1, the approach presented in this work is aimed at AAL and healthcare applications. The ability of such systems to predict future activities of occupants allow the implementation of multiple valuable services. In particular, it allows the detection of deviations from standard routines (e.g. the occupant does not perform the activity that we predicted they would do), which may indicate degradation in health or well-being. It also allows to provide anticipatory services which can remind the occupant about certain health-related events (e.g. remind the occupant that they will soon need to take their medication, if we predicted that this is the next activity they should do based on their past behaviour).

4.2.3 Machine learning

In Section 4.2.2, we discussed works that use machine learning algorithms for prediction, after a preliminary sequence mining step. This sequence mining step is typically used to extract a number of features (such as sensor use per time period).

However, modelling human routines in homes from a theoretical standpoint, especially when there is such variability in smart home environments (different sensor installations, different home layouts, etc.). As such, it is not guaranteed that

hand-crafted features, extracted using sequence mining, will actually be valuable to model activity prediction for all homes, let alone be optimal.

Therefore, we find several works that directly use machine learning techniques, without any preliminary sequence matching. Most works are based on graphical models (such as HMMs) as in [11, 94, 74, 104, 118, 164]. We do find some examples of deep learning approaches (such as Long Short-Term Memory neural networks (LSTMs)) applied to activity prediction, as in [71, 31, 76]. We discuss in more details 2 papers that present graphical prediction approaches in Section 4.2.3.1 and Section 4.2.3.2.

4.2.3.1 Anticipatory temporal conditional random fields

In [79], Koppula and Saxena propose an activity prediction scheme from video data. Applications targeted by this work revolve around robotics, rather than smart home systems. The goal in such use cases is to anticipate, using visual data that could come from an autonomous robot, the future actions of a person in order to adapt its interactions, responses, and understanding of the environment.

The approach proposed by the authors consists in using human pose and surrounding objects to infer future actions. More precisely, a first step seeks to infer the functionality of objects in the frame of view, depending on how the object is interacted with (e.g. an object might be "drinkable" if it is often found near a human's mouth). By doing so, we can obtain a heatmap of object functionalities for a specific viewpoint, by observing the possible positions of such objects over time.

In a second step, the current and past situations are modelled using à CRF, which integrates variables representing human poses, object functionalities, object locations, and sub-activities (or tasks, as we call them in our thesis). An anticipatory temporal CRF is then generated to model a possible future situation, by extending the previous CRF with trajectories and future poses, functionalities, locations, and sub-activities. In order to predict properly, the system generates as many anticipatory temporal CRFs as there are possible future situations. The most probable one is then selected as the prediction the system makes about the future activity.

Contrary to previous activity prediction methods presented in Section 4.2, the approach of Koppula and Saxena is designed to consider every possible future situation before actually predicting the most likely one. While there may be some situations where this would be computationally prohibitive, such an approach might be applicable in smart home environments, in which knowledge about the home and the occupants can help eliminate logically impossible scenarios, and thus avoid illogical activity prediction. For example, certain activities may be only physically performable in certain places (an assumption which we exploited in our place-based approach in Chapter 3); we can therefore eliminate certain possible future situations depending on the place in which the occupant is predicted to be.

4.2.3.2 CRAFFT DYNAMIC BAYESIAN NETWORK

Nazerfard and Cook present in [111] an activity prediction approach based on DBNs, called CuRrent Activity and Features to predict next FeaTures (CRAFFT). In this model, 4 input variables are used to predict the future activity: the current activity, the current place where the occupant is situated, the current time of day, and the current day of week. These variables are direct representations of 3 of the primary context dimensions in the home, as we defined them in Section 2.1.2.1: activity, place, and time. As such, their approach uses purely contextual data, and does not directly use sensor data for activity prediction.

The particularity of CRAFFT's structure is that the future activity is not directly predicted from these 4 contextual variables. In fact, the future place, the future time of day, and the future day of week of the next activity are predicted from these 4 variables. Then, the future activity is predicted using these 3 new inputs in addition to the initial 4 observations. Nazerfard and Cook, through an experimental study, argue that this particular 2-step DBN architecture leads to better activity prediction performance than a more naïve direct prediction from the 4 context variables.

In a sense, this approach can be seen as a primary context prediction approach (excluding identity), which is more general than activity prediction. Indeed, by predicting future place and time information before predicting the future activity, the DBN actually anticipates a more complete situation than just the activity.

4.2.4 DISCUSSION

We have seen in this section that both sequence matching and machine learning techniques have been successfully used for activity prediction in smart homes. In some works, a combination of both methods has also been proposed.

The large corpus of available work on sequence matching techniques, compression algorithms, itemsets mining, etc., can be transferred on the problem of activity prediction in smart homes. We generally have a good understanding of the behaviour and complexities of such techniques, which can offer some guarantees on a prediction system based on such algorithms (e.g. in terms of running time). On the other hand, these approaches are generally fairly rigid, in that they aim at finding temporal relationships in data regardless of the actual environment in which these data were recorded. As such, it is difficult to adapt such algorithms to take into account some *a priori* knowledge about smart homes, as we had done in our place-based activity recognition approach (e.g. known sets of possible places in the home, location of sensors, etc.). Therefore, some works use sequence matching in order to construct some features from data, which is then used to train a machine learning algorithm that performs the actual activity prediction step.

In approaches based on machine learning, it seems that most techniques used rely on graphical models such as CRFs or DBNs. These algorithms typically

include explicit temporal relationships between elements of the model, which makes them well-suited for prediction problems, compared to other approaches such as MLPs or SVMs, where temporality is not modelled.

The CRAFFT prediction algorithm of Nazerfard and Cook, presented in Section 4.2.3.2, presents a particularly interesting approach. Their predictive architecture uses the primary context that we had identified in Section 2.1.2.1, in the form of activity, place, time of day, and day of week variable nodes in a DBN. In addition, the use of a DBN makes CRAFFT an especially attractive approach to use as a basis on which to experiment various improvements for activity prediction. Indeed, one can extend CRAFFT through the introduction of new variable nodes and edges between variables, based on the specificities of smart home environments. In particular, one can extend this model with nodes that represent other context dimensions such as availability, and therefore propose a model that provides context prediction in general, rather than just activity prediction. Such context prediction can serve a number of context-aware services such as our communication assistant service based on availability, which we will discuss in Chapter 5.

Our contributions to the problem of activity prediction, presented in Section 4.3, are thus based on CRAFFT. We will present a number of different extensions of CRAFFT, based on smart home heuristics, which should improve prediction accuracy compared to the original CRAFFT model.

4.3 Context-based activity prediction

In the following section, we present our contributions to the problem of activity prediction in smart homes, as defined in Section 4.1. We first introduce the algorithmic basis for our contributions, through the predictive model of Nazerfard and Cook [111] which we had identified in Section 4.2. Then, we present 4 direct contributions to this initial model: in Section 4.3.2.1, we discuss the use of sensor data for activity prediction; in Section 4.3.2.2, we argue on using non-Markovian prediction models; in Section 4.3.2.3, we propose to model the cognitive state of the occupant to improve prediction accuracy; in Section 4.4.5, we expose a complete prediction model that combines the 3 previous contributions. We conclude this section with a presentation, in Section 4.3.3, of the results we expect to observe in activity prediction experiments.

4.3.1 The CRAFFT dynamic bayesian network for activity prediction

In this section, we discuss the basis for our contributions on the problem of activity prediction. In Section 4.3.1.1, we give a concise introduction to DBNs, which we will use as our prediction algorithm. In Section 4.3.1.2, we present in more details the activity prediction scheme used by Nazerfard and Cook in [111]

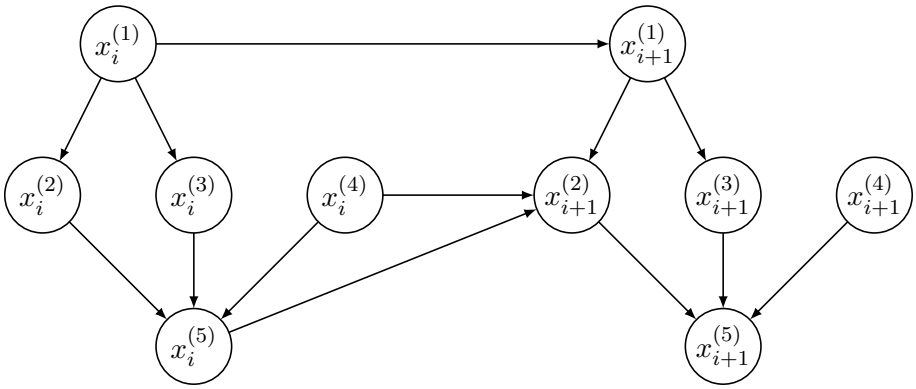

Figure 4.2 – A DBN of 5 variables between timesteps i and $i+1$.

(mentioned in Section 4.2). Their DBN architecture for activity prediction, called CRAFFT, will constitute the basis for our own contributions.

4.3.1.1 Dynamic bayesian networks

DBNs are an extension of BN (which were presented for activity recognition in Section 3.3.4.3) in which temporal dependences between variables from one timestep (sometimes called timeslice) to the following timestep can be represented. DBNs are thus suitable for prediction problems.

More formally, DBNs respect the homogeneous first-order Markov property, that is, dependencies between variables can only exist [69] inside a timestep (like a normal BN) or between two immediately consecutive timesteps. DBNs are called dynamic because they model the temporal evolution of variables; the topology of the graph itself does not change over time. As such, a DBN can be seen as a couple of two BNs, one which describes the initial distribution of variables (that is, the distribution of variables at time 0), and one which describes the transitions between two timesteps (which will be independent of time because of the homogeneous first-order Markov property).

DBNs possess two different types of parameters: the conditional probabilities of transition of variables, and the topology of the network itself [100]. The same techniques can be used in both DBNs and BNs to learn conditional probabilities from a training set. Expectation-Maximization or related gradient descent algorithms can be used to compute the initial conditional probabilities in a DBN, taking into account that these parameters must be tied between timesteps (a concept that does not exist in regular DBNs) [107]. Much like regular BNs, automatically learning the optimal topology of a DBN is a difficult problem.

Let us take the example of the DBN represented in Figure 4.2. Let $X_i = \{x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, x_i^{(4)}, x_i^{(5)}\}$ be the set of variables of this DBN at timestep i, and let $\mathrm{Pa}(x_i^{(j)})$ be the set of parents of variable $x_i^{(j)}$. We can then describe the joint

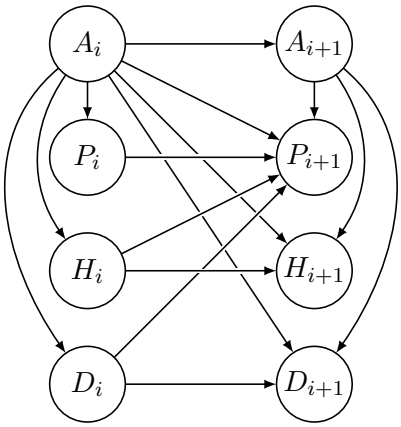

Figure 4.3 – Topology of the CRAFFT DBN between timesteps i and $i+1$ as reported in [111].

probability $\mathrm{p}(X_{i+1} \mid X_i)$ from one timestep to the next as follows:

$$\begin{aligned}\mathrm{p}(X_{i+1} \mid X_i) &= \prod_{x_{i+1} \in X_{i+1}} \mathrm{p}(x_{i+1} \mid \mathrm{Pa}(x_{i+1})) \\ &= \mathrm{p}(x_{i+1}^{(1)} \mid x_i^{(1)}) \\ &\quad \times \mathrm{p}(x_{i+1}^{(2)} \mid x_i^{(4)}, x_i^{(5)}, x_{i+1}^{(1)}) \\ &\quad \times \mathrm{p}(x_{i+1}^{(3)} \mid x_{i+1}^{(1)}) \\ &\quad \times \mathrm{p}(x_{i+1}^{(4)}) \\ &\quad \times \mathrm{p}(x_{i+1}^{(5)} \mid x_{i+1}^{(2)}, x_{i+1}^{(3)}, x_{i+1}^{(4)}).\end{aligned} \tag{4.1}$$

4.3.1.2 The CRAFFT and CEFA dynamic bayesian networks

Nazerfard and Cook introduce in [111] the CRAFFT DBN topology for activity prediction, which we illustrate on Figure 4.3. Four variable nodes are introduced in this DBN architecture:

— A_i, the Activity class performed at timestep i;

— P_i, the Place where the activity occurs at timestep i;

— H_i, the Hour of the day when the activity occurs. These values are discretized into 6 ranges of hours: $[0, 3]$, $[4, 7]$, $[8, 11]$, $[12, 15]$, $[16, 19]$, and $[20, 23]$;

— D_i, the Day of the week when the activity occurs, from 1 (Monday) to 7 (Sunday).

These 4 variables directly correspond to 3 of the primary context dimensions in the home: activity, place, and time. The CRAFFT model thus only uses

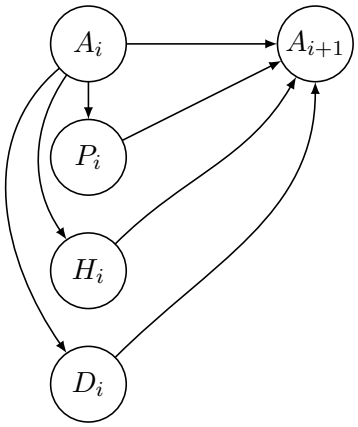

Figure 4.4 – Topology of the CEFA DBN between timesteps i and $i+1$ as reported in [111].

primary context information to predict future activities. Sensor data are thus ignored in this prediction model.

The choice of edges between nodes in the same timestep and edges between two consecutive timesteps relies on empirical observations by Nazerfard and Cook on smart home datasets. Examples of situations are given to justify, for example, the introduction of edges between H_i and P_{i+1} or D_i and P_{i+1}.

The particularity of the CRAFFT DBN is that the future activity A_{i+1} is not directly predicted from the observed variables (A_i, P_i, H_i, and D_i). Instead, a first step consists in predicting the contextual features of the next activity P_{i+1}, H_{i+1}, and D_{i+1}, which are then used in conjunction with A_i to predict A_{i+1}.

A simpler but more intuitive DBN structure for activity prediction that Nazerfard and Cook present in [111] is the CurrEnt Features and activity to predict the next Activity (CEFA) DBN, which we illustrate on Figure 4.4. While this architecture contains the same primary context variables, the future activity A_{i+1} is directly predicted from the currently observed features A_i, P_i, H_i, and D_i.

In their experiments, Nazerfard and Cook show that the CRAFFT DBN is significantly more accurate for activity prediction than the CEFA DBN, on 3 datasets of activities in smart homes. They also show that the CRAFFT DBN is more accurate than standard classifiers such as SVMs and MLPs used as predictors. In Section 4.4.1, we reproduce these experiments on 5 new datasets of activities in smart homes to see if the same observations can be made.

4.3.2 Beyond CRAFFT: PSINES and intermediate models

In this section, we present 4 contributions to the problem of activity prediction. Starting from the CRAFFT DBN, we discuss ways to improve the prediction accuracy of this predictor, based on the specificities of activities of

occupants in homes.

In Section 4.3.2.1, we propose to extend the CRAFFT architecture by introducing variables related to sensor data. We hope that valuable information for activity prediction can be found in such data sources, and that as such, discarding them as in the standard CRAFFT model is detrimental to prediction performance.

In Section 4.3.2.2, we propose to extend the influence of previous activities on the future activity by introducing additional edges in the CRAFFT DBN. By doing so, we hope to circumvent the Markovianness of activity sequences forced in CRAFFT, which intuitively is not a property held by daily living routines in homes.

In Section 4.3.2.3, we propose to introduce additional nodes related to the cognitive state of the occupant. We argue that such a variable has a great impact on the choices of activities the occupant decides to perfom, and that such a new node should thus improve prediction accuracy.

Finally, in Section 4.4.5, we present PSINES, a combination of all 3 previous propositions into one predictive DBN.

4.3.2.1 SCRAFFT: SENSOR-ENHANCED PREDICTION

In the standard CRAFFT topology, the 4 variables used to predict future activities only inform the model about the previous activity, as well as place and time information. While these data points indeed seem essential, as part of the primary context, for activity prediction, they do not expose much of the specificities of realization of activities by the occupant. Indeed, any variation in behaviour by the occupant, which may suggest particular routine patterns, or indicate changes in activity sequences, will be hardly detectable using only these 4 data points. While some instances could be indicative of such variations (for example, when the hour of the day at which the occupant performs an activity indicates a branching from routine to another), it ultimately seems that more data would be needed to retrieve such patterns of change.

In activity recognition, much of our input data comes from sensors installed throughout the home. In CRAFFT, such sensor data are not part of the DBN topology. Yet, the information of current activity (and possibly current place, at least in our place-based activity recognition approach), which are main variables of the CRAFFT model, are typically inferred from these sensor data, as we have seen in Chapter 3. As such, additional information about sensors can be introduced in the CRAFFT topology at no cost, since these sensors are already necessary for the activity recognition step.

On one hand, introducing sensor data in the CRAFFT model could inform the predictor about way the occupant performs the activity, which may indicate which activity they will do next. On the other hand, the 4 variables of the standard CRAFFT DBN intuitively should be the primary sources of information used to predict activities. As such, if one variable was introduced for each installed sensor, we might drown the more important data points into noise.

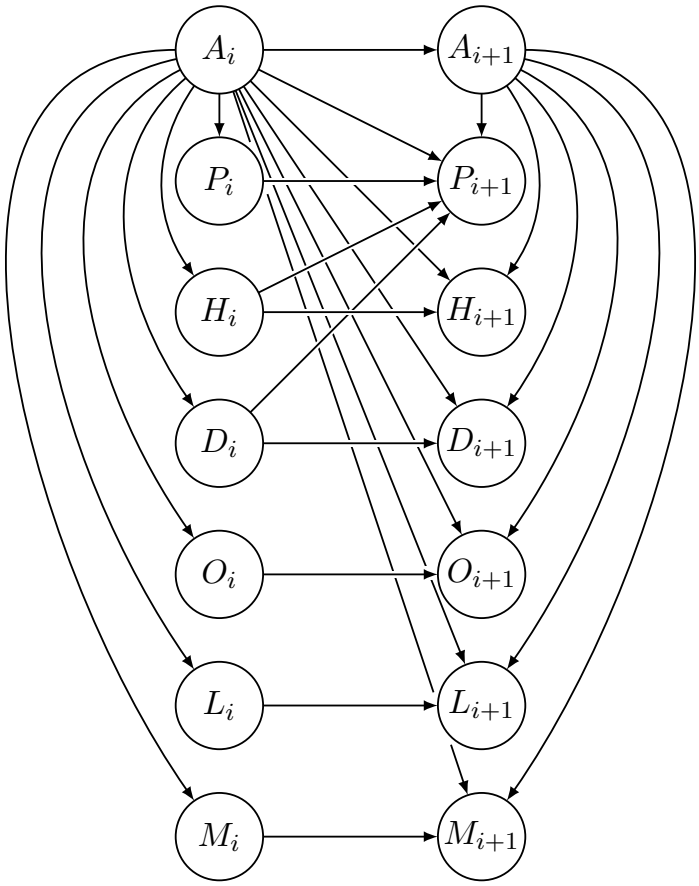

Figure 4.5 – Topology of the SCRAFFT DBN between timesteps i and $i+1$.

Therefore, we propose the Sensors CRAFFT (SCRAFFT) DBN, where new nodes are introduced to represent sensor aggregates. We illustrate a certain version of SCRAFFT on Figure 4.5. In this particular version of SCRAFFT, we find 3 new variables, compared to CRAFFT:

— O_i, the number of Openings of doors, cupboards, etc., during the activity at timestep i;

— L_i, the number of Light events recorded during the activity at timestep i;

— M_i, the number of Motion events recorded during the activity at timestep i.

Other version of SCRAFFT could be proposed with different sensor-related variables, or different aggregation techniques other than counting the number of events. The architecture we present here is influenced by the types of sensors available in the datasets we study in our experiments, presented in Section 4.4. We propose to quantize each sensor variable into 3 categories: "low", "medium", and "high", depending on the number of events recorded.

Similarly to place, hour of the day, and day of the week, these 3 sensor aggregate nodes influence their own future value at the next timestep, and are conditionally dependent on the activity node. The same 2-step prediction scheme used in CRAFFT is used in SCRAFFT, with O_{i+1}, L_{i+1}, and M_{i+1} being predicted along with P_{i+1}, H_{i+1}, and D_{i+1} before predicting A_{i+1}.

4.3.2.2 NMCRAFFT: NON-MARKOVIAN PREDICTION

Standard DBNs respect the first-order Markov property, as discussed in Section 4.3.1.1. As such, the CRAFFT model, which is a DBN, respects the same property. Therefore, in that predictive model, the future activity to predict is only influenced by the immediately preceding timestep. In particular, it is only influenced by the immediately preceding activity performed, and not by any other previous activities.

Intuitively, routines of daily living of occupants should not respect the first-order Markov property. Let us take the example of the standard day routine in Orange4Home (presented in Section 2.3.2.2 on Figure 2.11). If we consider the activity "Going up", we see that multiple different activities can occur after it: "Showering", "Computing", and "Using the sink". As such, it will be difficult to predict which activity comes after "Going up" if the predictor cannot access previous information. While other context information such as the hour of the day, used in CRAFFT, could disambiguate situations in some cases, it may be too limited in general (e.g. "Going up" then "Showering" and "Going up" then "Computing" both occur in the early morning in Orange4Home). We expect such non-Markovian situations to be common place in home routines.

On the other hand, we also expect that overfeeding the predictor with past information may decrease its performance. Indeed, the influence of past activities on the future activity should intuitively decrease the farther we go back in time. For example, activity information from previous days might not be very relevant to predict activities from a subsequent day, and might actually confuse the prediction algorithm.

Therefore, we propose to modify the CRAFFT DBN in order to circumvent the limitations induced by respecting the first-order Markov property. We thus introduce the Non-Markovian CRAFFT (NMCRAFFT) topology for activity prediction, in which the future activity depends not only on the immediately preceding activity, but on the last d activities that occurred. We call d the non-Markovian depth of the d-NMCRAFFT model.

We illustrate the 3-NMCRAFFT structure on Figure 4.6. We see that A_{i+1} is conditionally dependent not only on A_i, but also on A_{i-1} and A_{i-2}, since the non-Markovian depth used here is 3. In addition, A_i is also dependent on A_{i-2}, for symmetry reasons. The dependences between each consecutive activities already existed in the CRAFFT topology.

More generally, in a d-NMCRAFFT DBN, the future activity A_{i+1} is conditionally dependent on the previous d activities, and a previous activity $A_j, j \in$

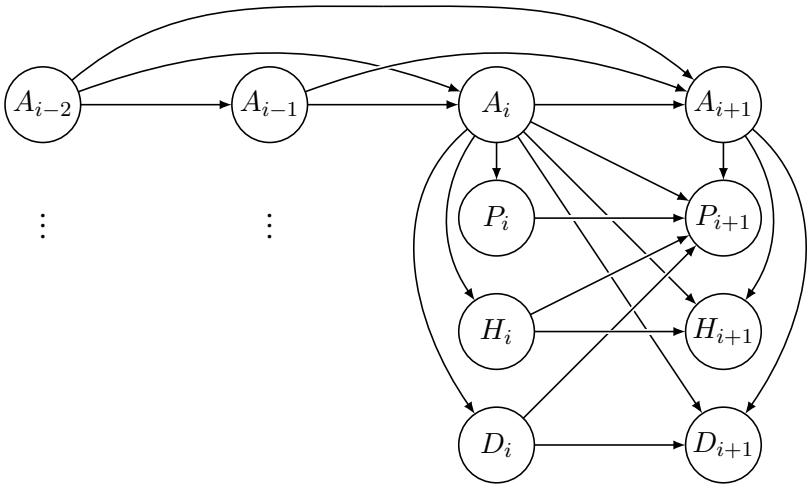

Figure 4.6 – Topology of the 3-NMCRAFFT DBN, between timesteps $i-2$ and $i+1$.

$[\![i-d+1, i]\!]$ is connected to the previous $j-(i-d+1)$ activities. In particular, this means that the 1-NMCRAFFT DBN is actually identical to the standard CRAFFT DBN.

Since regular DBNs respect the first-order Markov property, and although some works seek to relax this constraint and allow non-Markovian edges as in [44], the NMCRAFFT structure typically cannot be implemented in most DBN libraries. A simple solution to circumvent this constraint consists in duplicating activity nodes: for the example of the 3-NMCRAFFT, nodes A_{i-2} and A_{i-1} can be duplicated in timestep i. In this timestep, the edges between these nodes and A_i and A_{i+1} can be introduced like in a standard DBN. Additional edges between timesteps i and $i+1$ are added to link activities from one timestep to the next (for example, the duplicated node A_{i-1} in timestep i is linked to the duplicated node A_{i-2} in timestep $i+1$).

4.3.2.3 CSCRAFFT: MODELLING A COGNITIVE STATE OF THE OCCUPANT

In previous sections, we proposed to improve the CRAFFT DBN by introducing new nodes and edges related to sensor data (in SCRAFFT) or past context data (in NMCRAFFT). In any of these 3 models, we seek to predict future activities through rather indirect relationships: for example, in NMCRAFFT, we state that the future activity depends on past activities, when such dependence is in fact merely statistical and not, in a sense, a relationship of causation. For example, the fact that an occupant is always "Eating" after "Cooking" does not mean that the activity "Cooking" causes activity "Eating".

In fact, the real cause of transitions from one activity to another is the occupant themselves: their will, their mood, their behaviour, etc., will ultimately

drive them to perform an activity. In many cases, they will perform activities in a logical sequence with no deviation from what we expect to be normal (such as "Eating" after "Cooking"). In these cases, sensor and context data will contain should be sufficient to accurately predict activities. However, when the occupant's mood or will has a noticeable effect in the activity they choose to perform (e.g. "Telephoning" after "Cooking", because they remembered that they have an important call to make), such sensor and context-based prediction strategies might not be as accurate.

Evidently, measuring the will, the mood, etc., of an occupant is much more problematic than acquiring smart home sensor data or than inferring context data such as activity. There are no convenient modalities currently to record brain activity (helmets used to capture electroencephalograms are prohibitively cumbersome and very costly), and even if such modalities existed, learning a model linking brain activity to actual activity production in the home seems significantly out of reach as of today.

Nevertheless, we do find in the literature some works that include, in their model, the mental state of a user [70, 87]. Most of these papers are related to HCI and robotics; to the best of our knowledge, no such work exists for activity prediction in smart homes. In particular, we find the works of Mihoub and Lefebvre in [101], in which they tackle the problem of feedback prediction for public presentations through the use of a DBN architecture that includes the *cognitive state* of the orator. In this study, they show that the inclusion, in the DBN, of a node representing the cognitive state of the speaker does increase the feedback prediction accuracy of the model. This node can be latent, that is, unobserved and thus predicted by the DBN; this thus alleviate the need for recording brain activity. They also propose to discover *a priori* the observations for this cognitive state node, by applying different clustering algorithms on the rest of the data sources, with varying number of clusters (and thus cognitive state classes).

Therefore, we propose the Cognitive State CRAFFT (CSCRAFFT) DBN, in which a node representing the cognitive state of the occupant is introduced in the standard CRAFFT structure. We illustrate CSCRAFFT on Figure 4.7. The cognitive state of the occupant C_i at timestep i influences the current and future activity A_i and A_{i+1}, as well as their future cognitive state C_{i+1}. This node can either be latent or observed following a clustering step, as in [101].

Naturally, regardless of the latent or observed status of the cognitive state node, we have no guarantees that this newly introduced variable will reify the actual cognitive state of the occupant (which is itself an ill-defined concept). Nevertheless, we hope that introduction of such a node with these specific dependencies to activities will capture additional information that relates consecutive activities together.

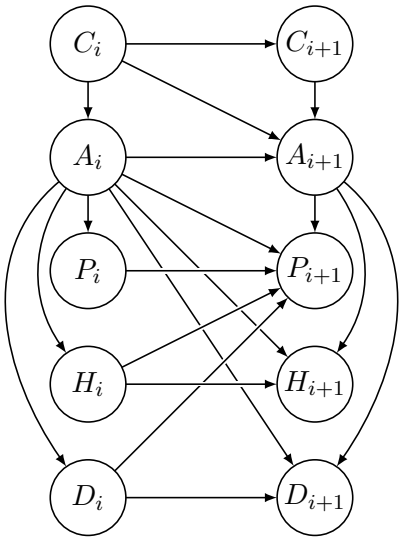

Figure 4.7 – Topology of the CSCRAFFT DBN between timesteps i and $i+1$.

4.3.2.4 PSINES

Our final contribution to the problem of activity prediction consists in a model which combines the contributions of the previous 3 proposed models: SCRAFFT, NMCRAFFT, and CSCRAFFT. This new model, which we call Past SItuations to predict the NExt Situation (PSINES), is thus a DBN whose topology is initially based on CRAFFT and that includes sensor aggregate nodes, non-Markovian edges between activities, as well as cognitive state nodes. We illustrate PSINES on Figure 4.8.

In this combined model, we extend the non-Markovian structure of edges between activity classes to the cognitive state nodes. Indeed, since these nodes directly influence activities, it seems natural that they follow the same non-Markovian pattern.

Partial combinations of models are also possible. For example, one can combine NMCRAFFT and CSCRAFFT, and not include sensor aggregate nodes from SCRAFFT. We will still refer to such partial combinations as PSINES. In experiments, different combinations of models can thus be used for different homes, depending on the improvements observed on each model separately from the others. For example, if the occupant of a particular home holds very erratic and changing routines, the inclusion of NMCRAFFT might be detrimental to prediction performance; we would thus not include it in PSINES for that particular home. For another home where the occupant follows very structured routines, NMCRAFFT should lead to better prediction accuracy; in that case, we would thus include it in PSINES.

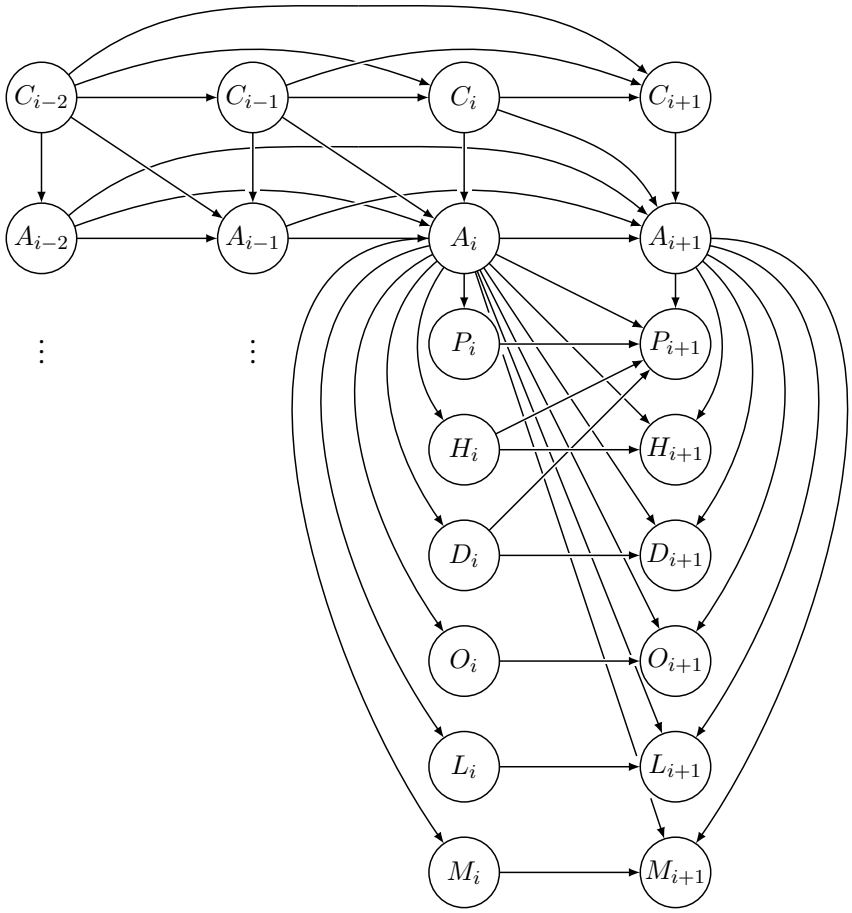

Figure 4.8 – Topology of the PSINES DBN between timesteps $i-2$ and $i+1$.

4.3.3 Expected results

We proposed in Section 4.3 a number of new DBN topologies for activity prediction that extend the CRAFFT model of Nazerfard and Cook [111]. The first of these new topologies, SCRAFFT, contains additional nodes that are directly linked to sensor data. We indeed expect that discarding all sensor information and exploiting only 4 high-level context information (activity, place, hour of the day, day of the week) is too limiting, and that some insight about the routine of the occupant, and thus their future activity, can be found in lower-level sensor data. In addition, we expect that this increase in input dimensionality will improve DBNs training, since it seems that there would be too few different possible input configurations using only the 4 context nodes originally in CRAFFT.

Therefore, we conjecture the following:

Hypothesis 4.1. *The SCRAFFT DBN, through the introduction of sensor aggregate nodes, will achieve higher activity prediction accuracy than the standard CRAFFT*

DBN.

The second of these new topologies, NMCRAFFT, was proposed after observing that routines of daily living should intuitively not follow the Markov property. In the CRAFFT DBN, only the last activity has influence on the future activity, which we expect does not reflect the reality of routines of occupants.

Therefore, we conjecture the following:

Hypothesis 4.2. *Routines of activities of occupants of homes do not have the Markov property. As such, the NMCRAFFT DBN will achieve higher activity prediction accuracy than the standard CRAFFT DBN.*

The third topology, CSCRAFFT, was proposed in order to explicitly include in the DBN model a representation of the cognitive state of the occupant. In particular, such a cognitive state node can be latent, that is unobserved, yet still influence the prediction of the DBN, which is essential as there are obviously no convenient and reliable method to objectively measure such cognitive information currently.

Therefore, we conjecture the following:

Hypothesis 4.3. *The CSCRAFFT DBN, through the introduction of a node representing the cognitive state of the occupant, will achieve higher activity prediction accuracy than the standard CRAFFT DBN.*

The final proposed topology, PSINES, is the result of combining the previous 3 topologies into one. We expect that the conjunction of all 3 contributions will model activity prediction more accurately than the initial CRAFFT DBN on which it is based.

Therefore, we finally conjecture the following:

Hypothesis 4.4. *PSINES will achieve the highest prediction accuracy of all DBN topologies previously mentioned.*

4.4 Experiments

In the following section, we experimentally study the behaviour of the various DBN models for activity prediction discussed in Section 4.3. In Section 4.4.1, we first examine the performances of the CEFA and CRAFFT model, in order to corroborate (or contradict) the claims made by Nazerfard and Cook in [111]. In Section 4.3.2.1, we study the performances of the SCRAFFT DBN, linked to Hypothesis 4.1. In Section 4.3.2.2, we evaluate the effect of non-Markovian depths on activity prediction through the NMCRAFFT DBN, which sheds light on 4.2. In Section 4.3.2.3, we experiment on the effect of introducing a cognitive state node on prediction accuracy, in relation with Hypothesis 4.3. Finally, in Section

4.4.5, we study the performances of the full PSINES model, in order to validate Hypothesis 4.4.

We used the implementation of DBNs provided by Murphy in the Bayes Net Toolbox for MATLAB [106].

For these experiments, we used 5 of the CASAS datasets (discussed in Section 2.3.1.2): HH102, HH103, HH104, HH105, and HH106, available on the CASAS project website [1]. Each of these 5 datasets contains the activities of only one occupant, along with the collected sensor data. These 5 datasets were all collected during the same time period (June to August 2011): as such, we can use the same training protocol for all datasets, and more meaningful comparisons from one dataset to the other can be made. The datasets respectively contain 30, 28, 32, 30, and 33 different activity classes (the vast majority of which are shared between datasets), such as "Sleep", "Bathe", or "Watch TV". In these datasets, we generally do not have information on the location of the occupant. As such, the place node in the tested models will always have a constant value of 1.

Much like in our activity recognition experiments on Orange4Home, we use a realistic temporal training protocol, where the first weeks of data are used as a training set and a validation set, and where the last weeks are used as the test set. On the CASAS datasets here, which all span the same time period, the training set spans from June 20, 2011 to July 7, 2011 (inclusive); the validation set spans from July 8, 2011 to July 17, 2011; the test set spans from July 18, 2011 to August 7, 2011.

For the last experiments in Section 4.4.5 on PSINES, we also used the Orange4Home dataset. In order to check the feasibility of our communication assistance service, we indeed need to evaluate the performances of an activity prediction approach on a dataset for which we can study availability prediction, which we discuss in Chapter 5.

4.4.1 Prediction performance with classical classifiers, CEFA, and CRAFFT

We report in Table 4.1 the prediction accuracy of the CRAFFT and CEFA models introduced in [111]. We also report in this table the prediction accuracy of a MLP, a SVM, and a BN, whose input vector contain the same features as the CRAFFT and CEFA models (previous activity, place, hour of the day, day of the week), as presented in Section 4.3.1.2.

Prediction performance seems to highly vary from one CASAS dataset to another: for example, the future activity is accurately labelled on average 47.69% of the time on the HH103 dataset, while we only reach an average accuracy of 17.10% on the HH102 dataset. This suggests that the inherent difficulty of each dataset is quite different among the 5 CASAS datasets we chose. This difficulty probably stems from the occupant themselves: it will be easier to predict future activities for occupants with highly regular routines compared to occupants with more erratic routines.

Nevertheless, this first set of experiments allows us to empirically verify the claims made by Nazerfard and Cook in [111]: we observe that, for all but the HH103 dataset, the CRAFFT DBN model is significantly more accurate than the CEFA DBN model (29.59% prediction accuracy on average compared to 26.36%). These results corroborate the findings of Nazerfard and Cook made on other datasets, that predicting future features before predicting the future activity is more accurate than directly predicting the future activity with DBNs.

However, contrary to the results in [111], the CRAFFT DBN is not always more accurate than other classical classifiers here. On average, the accuracy of CRAFFT (29.59%) lies, in our experiments on the CASAS datasets, between the accuracy of a MLP (28.51%) and the accuracy of a SVM (31.39%). Therefore, we cannot confirm using our experiments that the CRAFFT DBN is significantly more accurate for activity prediction in smart homes compared to other prediction techniques that rely on classical classifiers.

All in all, we see that the prediction accuracy on any of the 5 tested datasets is quite low. The best predictor on the easiest dataset (the SVM on HH103) only reaches a prediction accuracy of 49.66%, which is an obviously unacceptable accuracy if we were to use such a system to provide context-aware services. Indeed, if we use our example communication assistance service, it would not be able to provide valuable information to people wishing to correspond with an occupant of the home: if the system is wrong about future activities (and thus availabilities) of the occupant, it cannot accurately indicate future times at which that occupant will be available. This first set of results indicates that the problem of activity prediction in smart homes is significantly more difficult than the problem of activity recognition.

Still, accuracies obtained here are significantly better than predicting uniformly at random the next activity: for a dataset of 30 classes (such as HH102 and HH105), such a random predictor would achieve an accuracy of 3.33%, which is significantly worse than all other predictors we tested.

Therefore, further work is required to greatly improve the prediction performance of such approaches. In the next experiments, we study the effect of some changes we can apply to the CRAFFT DBN architecture so as to improve its accuracy.

4.4.2 Sensor-enhanced prediction

We report in Table 4.2 the prediction accuracy of the SCRAFFT model (presented in Section 4.3.2.1) on the CASAS datasets. We also included in this table the prediction accuracy of the original CRAFFT model (which were already reported in Table 4.1) for comparison. Quantization bounds for the 3 sensor aggregates (Openings, Lights, and Motion) for each datasets were obtained by applying the k-means algorithm to obtain 3 clusters (corresponding to the 3 quantized values “low”, “medium”, and “high”) on the training part of the dataset.

We can see that the SCRAFFT DBN is significantly more accurate than the

Predictor	Dataset					Average
	HH102	HH103	HH104	HH105	HH106	
MLP	17.74%[1]	49.24%[1]	31.91%[2]	18.29%[3]	25.36%[1]	28.51%
SVM	20.44%[4]	49.66%[5]	33.21%[6]	27.24%[6]	26.38%[7]	31.39%
BN	10.03%	45.25%	28.56%	11.75%	23.11%	23.74%
CEFA	16.20%	48.16%	28.67%	18.28%	20.47%	26.36%
CRAFFT	21.11%	46.13%	30.96%	25.22%	24.54%	29.59%
Average	17.10%	47.69%	30.66%	20.16%	23.97%	27.91%

Parameters:

[1] 150 hidden neurons, 100 epochs, 0.2 learning rate, 0.1 momentum.

[2] 200 hidden neurons, 100 epochs, 0.2 learning rate, 0.1 momentum.

[3] 175 hidden neurons, 100 epochs, 0.2 learning rate, 0.1 momentum.

[4] $C = 10000$, $\gamma = 0.001$.

[5] $C = 100$, $\gamma = 0.01$.

[6] $C = 1000$, $\gamma = 0.001$.

[7] $C = 10$, $\gamma = 0.01$.

Table 4.1 – Prediction accuracy of the CRAFFT and CEFA DBNs described in [111], as well as MLP, SVM, and BN using the same input data.

CRAFFT DBN on all but the HH102 dataset. For example, on the HH103 dataset, the accuracy gap is greater than 4% (50.25% compared to 46.13%).

This suggests that, in the majority of cases, sensor data, even when aggregated, brings valuable information to predict future activities, as anticipated in Hypothesis 4.1. As such, the task of activity prediction may not be optimally solved when we discard all sensor data, as is done in the original CRAFFT model. On the other hand, we see that introducing these sensor aggregate nodes in the DBN topology actually degrades performance on the HH102 dataset. There are thus cases where using sensor data would actually be detrimental to accurate activity prediction.

Based on previous observations, we can hypothesize that predicting future activities directly from raw sensor data may lead to even more accurate predictions (as aggregation does potentially remove salient information). DBNs are not the most well-adapted methods to process raw sensor data; other algorithms, such as LSTMs, may be used for activity prediction from raw data instead. These techniques usually require a large training set to be accurate, as the feature space when working on raw data is much larger than when working on higher-level context information (which is the case in SCRAFFT). As we discussed in Chapter 3, obtaining extensive training datasets in smart homes aimed at the general public is largely unrealistic.

While additional sensor data seems to improve activity prediction, the perfor-

Model	Dataset					Average
	HH102	HH103	HH104	HH105	HH106	
CRAFFT	**21.11%**	46.13%	30.96%	25.22%	24.54%	29.59%
SCRAFFT	20.00%[1]	**50.25%**[2]	**31.66%**[3]	**26.45%**[4]	**27.62%**[5]	**31.20%**

Quantization bounds (x, y), low $< x \leqslant$ medium $\leqslant y <$ high:

[1] Openings: $(2, 5)$, Lights: $(54, 195)$, Motion: $(89, 292)$.

[2] Openings: $(1, 1)$, Lights: $(20, 65)$, Motion: $(32, 125)$.

[3] Openings: $(4, 13)$, Lights: $(39, 137)$, Motion: $(102, 459)$.

[4] Openings: $(2, 6)$, Lights: $(58, 240)$, Motion: $(54, 192)$.

[5] Openings: $(2, 3)$, Lights: $(41, 170)$, Motion: $(77, 319)$.

Table 4.2 – Prediction accuracy of the CRAFFT and SCRAFFT DBNs on the CASAS datasets.

mances of the original CRAFFT model are sufficiently close to the SCRAFFT model to conclude that the majority of information used to predict activities comes from the 4 context nodes used in CRAFFT: activity, place, hour of the day, and day of the week. Predicting activities exclusively from raw sensor may thus not be as accurate as the SCRAFFT approach. Further work on predicting activities from both context dimensions and raw sensor data may shed some light on which data sources are required for optimal activity prediction.

4.4.3 Non-Markovian prediction

We report in Table 4.3 the prediction accuracy of the NMCRAFFT model (presented in Section 4.3.2.2) with varying non-Markovian depth on the CASAS datasets. We provide in Figure 4.9 a graphical representation of these results. We can observe 3 different trends between non-Markovian depth and prediction accuracy, depending on the dataset:

1. Prediction accuracy decreases when non-Markovian depth increases. This trend is observed on HH102. The best result (21.11% for HH102) is thus obtained with the 1-NMCRAFFT model, that is, the original CRAFFT model presented in [111].
2. Prediction accuracy increases and reaches its peak at a non-Markovian depth of 2, and then decreases for larger depths. This trend is observed on HH104 and HH105. The best results (31.60% for HH104 and 25.48% for HH105) are thus obtained with the 2-NMCRAFFT model, when the future activity class conditionally depends on the previous 2 activity classes.
3. Prediction accuracy increases and reaches its peak at a non-Markovian depth of 3, and then decreases for larger depths. This trend is observed on HH103 and HH106. The best results (51.08% for HH103 and 27.75% for HH106) are thus obtained with the 3-NMCRAFFT model, when the future activity

	Dataset					
Depth	HH102	HH103	HH104	HH105	HH106	Average
1	**21.11%**	46.13%	30.96%	25.22%	24.54%	29.59%
2	19.27%	49.81%	**31.60%**	**25.48%**	26.29%	30.49%
3	18.72%	**51.08%**	30.27%	24.71%	**27.75%**	**30.51%**
4	18.72%	50.96%	29.66%	24.59%	25.71%	29.93%

Table 4.3 – Prediction accuracy of the NMCRAFFT DBN with varying non-Markovian depth on the CASAS datasets.

class conditionally depends on the previous 3 activity classes.

We see in these trends that in 4 out of 5 datasets, non-Markovian models perform better than the Markovian CRAFFT approach. Indeed, prediction accuracy increases on all datasets but HH102 when using a 2-NMCRAFFT model compared to CRAFFT. Trend 3 shows that for some datasets (HH103 and HH106), prediction accuracy is even higher with a 3-NMCRAFFT model. However, the 4-NMCRAFFT model is less accurate for all 5 datasets, compared to the 5-NMCRAFFT model.

These observations are well condensed in the average results we obtain on the CASAS datasets: prediction accuracies for the 2-NMCRAFFT model (30.49%) and the 3-NMCRAFFT model(30.51%) are nearly identical, while prediction accuracies for the CRAFFT model (29.59%) and the 4-NMCRAFFT model (29.93%) are worse.

These results suggest that activity sequences do not, on average, have the Markov property, as we anticipated in Hypothesis 4.2. However, we observed that the home and occupants monitored by the system have a great impact on this result. In particular, it seems that for some homes (such as in the HH102 dataset), the routine of the occupant is for the most part a Markov process, contradicting Hypothesis 4.2. In other homes, the ideal non-Markovian depth to use for activity prediction can vary, although depths of 4 and more seem detrimental to prediction accuracy on the CASAS datasets. Therefore, estimating the optimal non-Markovian depth for a specific home is essential, and should be the focus of future work in activity prediction for smart homes (not addressed in this thesis).

4.4.4 Cognitive states for prediction

We report in Table 4.4 the accuracy of the pre-clustered CSCRAFFT model (presented in Section 4.3.2.3) on the CASAS datasets, where the cognitive state node is pre-labelled using the k-means clustering algorithm. We performed the experiments with 5, 10, and 20 clusters (i.e. possible states for the cognitive state node).

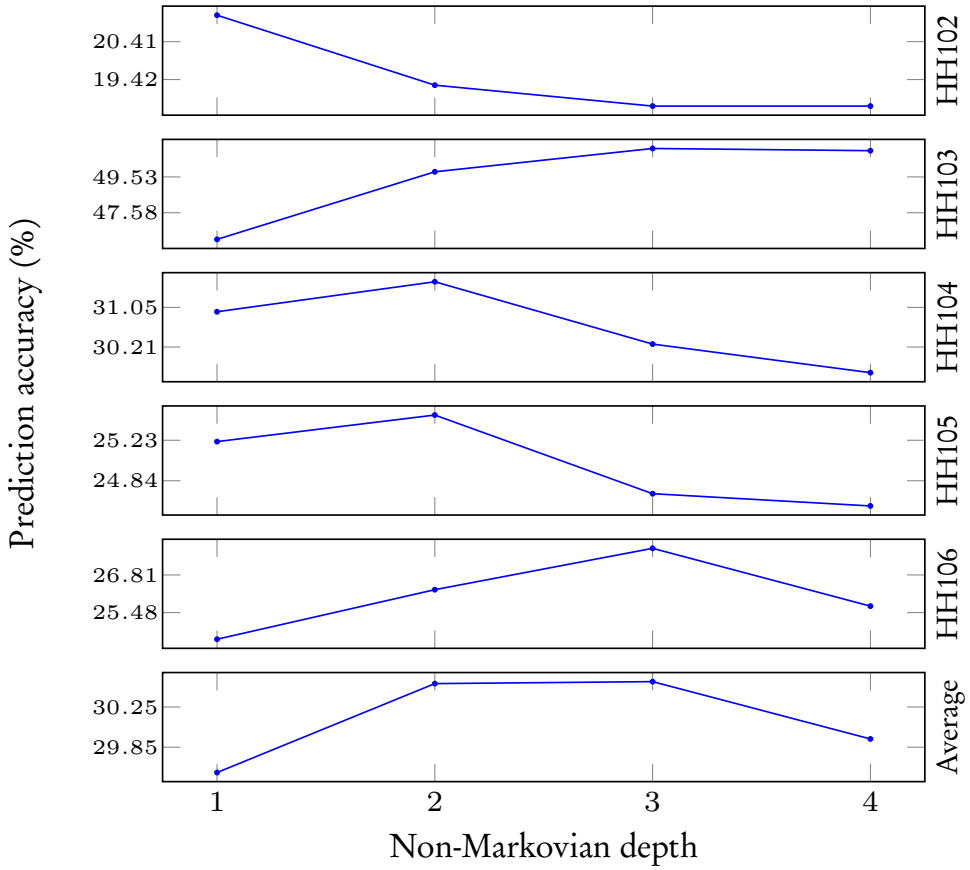

Figure 4.9 – Prediction accuracy of the NMCRAFFT DBN with varying non-Markovian depth on the CASAS datasets.

We observe that the pre-clustered CSCRAFFT DBN is approximately as accurate as the CRAFFT DBN, or even slightly worse than the CRAFFT DBN depending on the number of states. For a number of states of 10, the pre-clustered CSCRAFFT model is a full percent less accurate than the CRAFFT model (28.43% compared to 29.59%).

We present on Figure 4.10 the prediction accuracy of the latent CSCRAFFT model (presented in Section 4.3.2.3) on the CASAS datasets, where the cognitive state node is unobserved. The accuracy of the original CRAFFT model is included in the figure for comparison. We performed the experiments with a varying number of possible states for the cognitive state node, from 2 to 20. We report in Table 4.5 the best, worst, and average prediction accuracy of CSCRAFFT on each dataset.

Much like for the NMCRAFFT model, we observe three different trends, depending on the dataset:

1. Prediction accuracy is mostly unaffected by the latent cognitive state node.

Number of states	Dataset					Average
	HH102	HH103	HH104	HH105	HH106	
5	**21.38%**	45.25%	**30.96%**	**25.22%**	24.44%	29.45%
10	20.02%	45.37%	29.46%	23.83%	23.47%	28.43%
20	21.02%	**46.26%**	30.34%	23.58%	23.76%	28.99%
CRAFFT	21.11%	46.13%	**30.96%**	**25.22%**	**24.54%**	**29.59%**

Table 4.4 – Prediction accuracy of the pre-clustered CSCRAFFT DBN on the CASAS datasets.

Model	Dataset				
	HH102	HH103	HH104	HH105	HH106
Best	22.47%	46.64%	30.96%	25.73%	24.73%
Worst	21.38%	45.50%	30.87%	25.22%	23.67%
Average	21.90%	45.99%	30.96%	25.49%	24.17%
CRAFFT	21.11%	46.13%	30.96%	25.22%	24.54%

Table 4.5 – Prediction accuracy of the CSCRAFFT DBN with unobserved latent nodes on the CASAS datasets.

This trend is observed on HH104.

2. Prediction accuracy of CSCRAFFT is on average better than CRAFFT. This trend is observed on HH102 and HH105.
3. Prediction accuracy of CSCRAFFT is on average worse than CRAFFT. This trend is observed on HH103 and HH106.

However, we can note that the best configuration of CSCRAFFT (i.e. the right number of states) is always better than CRAFFT for each dataset (except for HH104 where they have identical accuracies). Hypothesis 4.3 is thus somewhat verified, although the improvements in prediction accuracy seem smaller in this case.

Indeed, the improvements in prediction accuracy with the latent CSCRAFFT DBN are quite smaller than those obtained with SCRAFFT or NMCRAFFT. While the theoretical motivation for introducing such a node are sound (the routine of the occupant will necessarily be influenced by their mental state, their thoughts, etc.), and while introducing such a node in a DBN model has lead to improvements in prediction accuracy in previous works of the literature [101], it does not seem to be as straightforwardly applicable to smart home situations. One possible explanation is that the relationship between the routine of the occupant

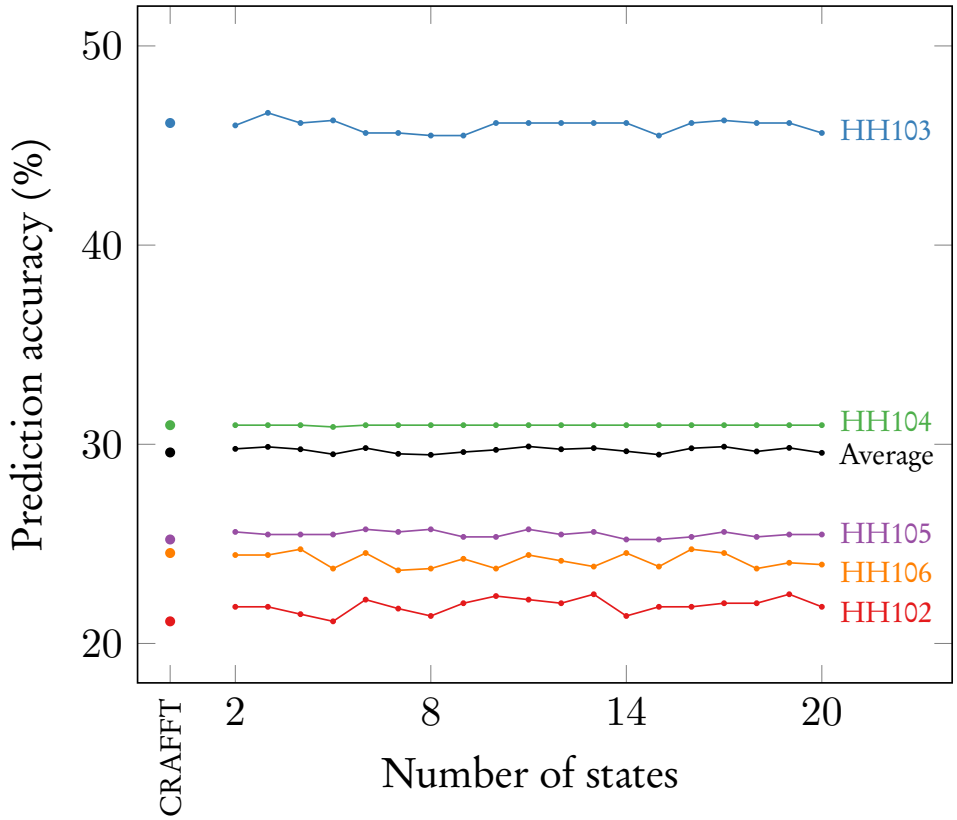

Figure 4.10 – Prediction accuracy of the latent CSCRAFFT DBN with varying number of states for the latent cognitive state node on the CASAS datasets.

and their cognitive state is much more complex than in other application domains (such as oral presentations as in [101]), such that no pattern for this relationship can be learned by the DBN from contextual data about the home.

4.4.5 Combined model for prediction

We report in Table 4.6 the prediction accuracy of the combined PSINES DBN on the CASAS datasets. We also report the previous best results for the CRAFFT, SCRAFFT, NMCRAFFT, and CSCRAFFT DBNs.

This final set of experiments shows that combining all 3 improvements (sensor aggregate nodes, non-Markovianness, and cognitive state nodes) into one model leads on average to a model that is more accurate than any other DBN predictor we have tested, as we anticipated in Hypothesis 4.4. For example, PSINES reaches a prediction accuracy of 52.03% on HH103, which is almost 1% higher than the accuracy of the second best DBN (3-NMCRAFFT with 51.08%), and nearly 6% higher than the initial CRAFFT DBN of Nazerfard and Cook (46.13%). PSINES performs worse than another tested DBN only on HH105 in our experiments,

Model	Dataset					Average
	HH102	HH103	HH104	HH105	HH106	
CRAFFT	21.11%	46.13%	30.96%	25.22%	24.54%	29.59%
SCRAFFT	20.00%	50.25%	31.66%	**26.45%**	27.62%	31.20%
NMCRAFFT	21.11%	51.08%	31.60%	25.48%	27.75%	30.51%
CSCRAFFT	**22.47%**	46.64%	30.96%	25.73%	24.73%	30.11%
PSINES	**22.47%**[1]	**52.03%**[2]	**32.19%**[3]	26.07%[4]	**28.49%**[5]	**32.25%**

Parameters:

[1] 1-NMCRAFFT + 13 latent states CSCRAFFT.

[2] SCRAFFT + 3-NMCRAFFT + 3 latent states CSCRAFFT.

[3] SCRAFFT + 2-NMCRAFFT + 2 latent states CSCRAFFT.

[4] SCRAFFT + 2-NMCRAFFT + 6 latent states CSCRAFFT.

[5] SCRAFFT + 3-NMCRAFFT + 16 latent states CSCRAFFT.

Table 4.6 – Prediction accuracy of the CRAFFT, SCRAFFT, NMCRAFFT, CSCRAFFT, and combined PSINES DBNs in their best configurations on the CASAS datasets.

Model	Orange4Home
CRAFFT	61.68%
2-NMCRAFFT	87.74%
3-NMCRAFFT	88.57%
CSCRAFFT	63.55%
PSINES	**89.52%**

Table 4.7 – Prediction accuracy of different models on the Orange4Home dataset.

where it still comes second after SCRAFFT (26.07% against 26.45%).

4.4.5.1 Orange4Home

We applied the PSINES DBN to the Orange4Home dataset, using, like in our activity recognition experiments, the first 2 weeks of data as a training set, the third week as validation set, and the last week as the test set. We present the results of this experiment in Table 4.7. PSINES reaches on this dataset a prediction accuracy of 89.52%. The standard CRAFFT model, on the other hand, only reaches an accuracy of 61.68%, which is substantially less.

The main improvements in prediction accuracy on this dataset come from the inclusion of non-Markovian previous activities. Indeed, 2-NMCRAFFT and 3-NMCRAFFT obtain prediction accuracies of 87.74% and 88.57% respectively, which is close to the performance of the full PSINES model. On the other hand, the inclusion of cognitive state nodes through CSCRAFFT only marginally improves accuracy: CSCRAFFT with 6 states by itself reaches a prediction

accuracy of 63.55%.

The importance of introducing non-Markovianness to the prediction model can be easily illustrated in the Orange4Home dataset. Let us take the example of the activity "Going up" in the Staircase. If we only use this activity as well as related context information of place, hour of the day, day of the week, as in CRAFFT, it is obviously very difficult to predict what the occupant will do because multiple rooms can be reached and thus multiple activities can be performed once the user changed floor (although hour of the day and day of the week might give some hints as to why the occupant was going upstairs). On the other hand, if the predictor knows the previous 2 or 3 activities that the occupant was doing before going upstairs, it will most likely have an easier time identifying a part of the routine of the occupant and thus predict accurately their next activity.

Most confusions thus occur when the occupant abruptly changes their routine in an unexpected manner. For example, PSINES incorrectly predicts that the occupant will perform "Preparing" in the Kitchen on Tuesday 21, 2017, when in fact the occupant decided to leave the home to have lunch. In another example, PSINES predicts twice, on Tuesday 21, 2017 and Wednesday 22, 2017, that the occupant will be "Watching TV" in the Office at the end of the day, when they in fact will be "Going down" directly because they decided to skip watching TV those two days. In both of those examples, these changes in routine had never occurred before in the training set, and thus could not be learned by PSINES.

All in all, we see that, using the PSINES model, we obtain a predictor that reaches fairly acceptable prediction accuracy on the Orange4Home dataset, which is a necessity for our communication assistance service. This service would not be implementable if we were using the standard CRAFFT approach, which does not predict activities sufficiently accurately.

4.5 Conclusions

We presented in this chapter the third contribution of our thesis: PSINES, a context-based activity prediction model. We experimentally evaluated this approach, as well as intermediary contributions, on 5 datasets of the literature. We showed that each partial contribution can improve prediction accuracy, and that PSINES is the most accurate model we tested on these datasets, on average. We obtained similar results on the Orange4Home dataset.

First, we showed that sensor data can be valuable for activity prediction, in addition to context data. These findings may influence the choice of sensor installations and prediction algorithms used in smart home systems that require activity prediction capabilities.

Second, we also showed that routines of daily living do not, generally, respect the first-order Markov property. As such, predictors that inherently are first-order Markov models (such as the original CRAFFT model) will be limited in

their prediction capabilities. Non-Markov models exhibit, in our experiments, significantly more accurate prediction performances.

Third, we showed that the introduction of a latent node, meant to represent the cognitive state of the occupant, can lead to slightly more accurate models. However, improvements are smaller than the other two contributions, and modelling (such as the optimal number of cognitive state classes) seems difficult. Further work is required to investigate the effects of introducing such latent nodes in DBN-based activity prediction models in smart homes.

Finally, we showed that the combination of these 3 contributions, PSINES, is the most accurate model we have studied for activity prediction in homes, according to our experiments. Ultimately, we saw that each specific dataset (and therefore, each specific home) requires a specific prediction model. Indeed, we have seen in our experiments that each of our contributions has significantly varying effects on activity prediction, due to the variety of different occupant behaviours and home environments. Therefore, we suggest that smart home systems which require activity prediction capabilities must construct their own prediction model independently of other homes, and not use a generic model that would be applied to any home.

Predicting future activities is the second necessary step to provide our example service of communication assistance to occupants. In order to provide such a service reliably, high prediction accuracy is essential. Our contributions to the problem of activity prediction are thus valuable for such a service. While prediction accuracy, despite our contributions, was still relatively low on the CASAS datasets, it can be considered acceptable on Orange4Home (89.52%), which is the only dataset, as far as we know, that also contains ground truth of availability for communication of an occupant. In Chapter 5, we thus study the problem of availability prediction, from previously recognized and predicted activities and related context information. We propose a new availability inference scheme that allows the implementation of such communication assistance services. We evaluate the accuracy of this availability inference model on the Orange4Home dataset. We discuss the implementation of such communication assistance services, based on our experimental results.

CHAPTER 5

Inferring availability using context

Context-aware smart home services require primary context information to operate properly. As such, activity recognition, studied in Chapter 3, and activity prediction, studied in Chapter 4, are necessary components of a smart home system. However, primary context might not always be sufficient to provide any context-aware service. For example, a communication assistant that manages incoming communications from outside the home will require secondary context information. In particular, it will require information about the availability of the home's occupants to be interrupted by an incoming communication. To illustrate the implementation of complex context-aware smart home services that require secondary context information, we thus study the problem of availability inference in this chapter. We properly define this problem and the assumptions we make in Section 5.1. Following a survey of state-of-the-art studies on availability inference, in Section 5.2, we present in Section 5.3 our contributions on availability inference, which propose to estimate availability directly from other context dimensions, and in particular from activities, correspondents, and modalities of communication. We experimentally evaluate this availability inference approach in Section 5.4.

5.1 Problem statement and preliminary assumptions

In the following section, we define more precisely what we mean by availability inference in Section 5.1.1. We state the assumptions we make about this problem in Section 5.1.2.

5.1.1 Availability inference

In this thesis, we define availability inference as the problem of estimating the current availability (sometimes called "interruptibility" in the literature) for communication of an occupant based on current context and sensor data collected in the home. By availability for communication, we mean the degree of acceptability for interrupting the occupant in their home activity due to an incoming communication attempt from outside the home. A similar view of the problem is to predict the likelihood that the occupant will answer an incoming communication given their current situation and past behaviours.

We thus do not consider the problem of estimating the availability for communications which occur inside the home, such as communications between different occupants of the home. Such situations require the recognition of simultaneous occupants' activities, identification of occupants, etc., which we did not address in this thesis either for the problem of activity recognition in Chapter 3 or the problem of activity prediction in Chapter 4.

We do not study the problem of directly predicting future availabilities of an occupant, which seems to be a more difficult problem. Here, we propose to tackle the problem of current availability estimation first. We can actually derive an availability prediction system given a current availability inference approach and a future situation prediction system. Indeed, one can first predict future situations, and then infer the availability of an occupant on these situations to obtain an availability prediction system. For example, if we assume that availability highly depends on activity, we can use our activity prediction contributions presented in Chapter 4 in conjunction with our contributions in availability inference in this chapter to construct an availability prediction approach. We do not experimentally study this combination in this thesis.

5.1.2 Assumptions

In this section, we state the main hypotheses we make about the problem of availability inference (as defined in Section 5.1.1). Most of these assumptions were already made in for activity recognition and prediction. Justifications for these assumptions can thus be found in Chapter 3 and Chapter 4.

5.1.2.1 Single-occupant situations

We assume that only one person occupies the home at all times (although that person can change). This assumption was already made for activity recognition and activity prediction.

5.1.2.2 A priori identification

We assume that the smart home already has the capability of identifying the occupant in the home. For availability estimation, we cannot realistically assume

that the availability of an occupant is independent of their identity.

5.1.2.3 Sequentiality of activities

We assume that activity instances necessarily follow one another sequentially, i.e. the single occupant cannot perform two activities in parallel. This assumption was already made for activity recognition and activity prediction.

5.2 State of the Art

5.2.1 Availability in professional environments

Professional environments are a prime example of application domains where estimating the availability of people for communication is valuable. Indeed, sharing availabilities between employees allows them to meet and communicate at opportune times, and reduce inappropriate interruptions that would decrease productivity [144]. Many classical methods and habits have been employed to signify availability in professional environments: sharing professional schedules, leaving one's office door closed when unavailable, changing one's availability status on the professional instant messaging system, etc. As such, offices and professional environments have been the main focus of research works on automatic availability estimation [151].

We present 3 papers related to availability estimation in professional environments in Section 5.2.1.1, Section 5.2.1.2, and Section 5.2.1.3.

5.2.1.1 Inferring availability from posture and computer usage

Tanaka et al. present in [144] an availability estimation approach for professional offices that relies on information on head posture and interactions on the computer. Indeed, based on previous literature studies, they note that static head postures tend to reflect the degree of engagement between a user and their work task. Furthermore, concentration on one's task might be captured by the relative stillness of the head.

Computer interactions are also used to help determine availability. Examples of monitored interactions include keystroke events, mouse events, or transitions between computer applications. Tanaka et al. suggest that head posture will help reduce availability inference error in cases where the user is doing a task that does not require their computer (in which case computer interactions are absent) or for computer tasks with similar interaction patterns yet different availability implications.

The availability inference approach proposed by Tanaka et al. consists in binarizing each of the previously mentioned data sources and then summing them. A number of thresholds (2 in their work, where availability can take 3 different values) is used to decide whether the user is available or not based on

this sum. Binarization of variables is performed using preliminary cluster analysis on training data, during which binarization thresholds can be found.

5.2.1.2 Communication modality recommendations

In [54], Fogarty et al. propose a new communication client for workplaces which suggests appropriate communication modalities depending on the availability of one's colleague. This system, called *MyVine*, is intended to replace instant messaging applications typically used in professional settings. These applications, despite usually providing rudimentary availability information, are typically not context-aware in the sense that they do not take into account all relevant context information about the user in order to infer availability. Instead, these systems typically rely on professional schedules and computer interaction (in the binary sense), which provide only a limited view of a user's potential availability.

Fogarty et al. propose to enhance such systems mainly through the analysis of speech: detecting whether a user is speaking or not can be an important indicator of their availability. Much like previous systems, computer activity and schedule information are also used. Door opening sensors are also included in the availability inference approach: intuitively, employees who keep their office doors closed are most likely not available.

MyVine can thus expose information about availability and appropriate communication modalities based on these data sources and simple rules. This work defends the idea that availability for communication necessarily has to be linked to specific communication modalities. In other words, users may have different availabilities depending on the communication modality used. In particular, the more unavailable a user seems to be, the more *MyVine* will tend to suggest asynchronous modalities (such as e-mails) rather than synchronous modalities (such as instant messaging or face-to-face meetings) to communicate.

5.2.1.3 Scheduling e-mail delivery based on availability

Kobayashi et al. propose in [78] an e-mail delivery system that aims at limiting inappropriate interruptions at work. The authors argue that e-mails delivery should be delayed when a user's availability is low, so as to not disrupt their focus on a potentially demanding task. However, e-mails should not be delayed for too long either, so that communication is not hindered either. Productivity could indeed still decrease if e-mails are delayed too much yet delivered at times where users are available.

User availability is in this case estimated solely from computer interaction data. Kobayashi et al. propose to segment computer interaction into to distinct cases: moments when the user is switching between applications, and moments of stable use of a single application. The first case corresponds to situations where the user will probably be available, while the second case corresponds to situations where the user will probably be unavailable. Multiple interaction indicators are

used, such as keyboard and mouse usage, currently active window, number of simultaneously open applications, etc.

Kobayashi et al. experimentally show that delaying the delivery of e-mails based on users' availability significantly decreases their feeling that they were interrupted by an e-mail. As such, this work is an example that a communication assistance service that manages and potentially delays incoming communication can in fact be beneficial to users. Applications of these principles to communication modalities other than e-mails, and in smart home environments, are yet to be investigated.

5.2.2 Availability on smart phones

The democratization of smart phones in recent years has changed the way people communication with each other. Smart phones have become the main communication device for most of the population, due to its portability as well as the diversity of communication modalities integrated (phone calls, instant messaging, e-mails, etc.). However, mobile notifications can be very disruptive [131]. As such, estimating the availability for communication of smart phone owners has become another focus point of research on availability estimation, in addition to professional environments.

Estimating availability for communication on a smart phone presents new challenges that do not typically exist in professional environments. First, the only ambient sensors available are those that are included on the smart phone; in professional environments, offices themselves can be instrumented to help observe the behaviour of users (and thus help infer their availability). Second, the environment itself evolves during the day and is usually not known, since the smart phone follows the movements of its owner; in professional environments, the environment is relatively unchanged from one timestep to another. Third, the behaviour of the users covers the entire day, and thus multiple different situations; in professional environments, users are here for work and their behaviours and activities will thus be relatively controlled.

We discuss 2 papers on availability estimation on smart phones in Section 5.2.2.1 and Section 5.2.2.2.

5.2.2.1 Reducing mobile disruption in face-to-face conversations

Mayer et al. present in [97] a new approach for reducing the impact of smart phone disruptions (in particular incoming phone calls) on a face-to-face conversation. In this work, Mayer et al. propose to use an eye tracking system to evaluate the level of investment of a user in the conversation. Much like for head posture in Section 5.2.1.1, they argue, based on past works, that gaze is an important indicator of a user's interest in a conversation, and thus an indicator of their availability for mobile phone disruptions.

In their study, Mayer et al. conduct a set of experiments where 2 persons

have a conversation, while one of them has a mobile phone equipped with their proposed approach. After the experiments, they conduct interviews to evaluate the level of disruptiveness felt by user, depending on various configurations of their approach. These interviews help them make recommendations on the design of interaction models (for example to dismiss or accept phone calls) that are less disruptive during conversations.

An obvious drawback of this approach is the use of an eye tracking system, which are generally cumbersome, and not always portable. We feel that such sensors should not be used to evaluate availability of users for these reasons, even though they might provide valuable insight on availability.

5.2.2.2 Inferring partial availability to answer notifications

In [152], Turner et al. propose a new approach to model the availability of smart phone owners to consume notifications. Contrary to previous works, which assume binary reactions to notifications, they propose a multi-step model of availability where a user can either give a null, partial, or complete response to a notification. Reachability corresponds to the user noticing a notification. Engageability indicates partial response to the notification, which can be abandoned midway if they decide that the notification is not worth being interrupted. Receptivity corresponds to the user completely answering a notification.

The authors propose to infer the reachability, engageability, and receptivity of users to notifications using a J48 decision tree. This decision tree is trained to infer these 3 elements from a dataset of various smart phone data sources, such as acceleration data, audio volume, orientation, or charging state. Experimental results suggest that partial availability to notifications constitute a significant portions of interruptions. As such, using a system that can infer such partial answers should significantly reduce the number of misclassifications, compared to previous systems that predict binary availability.

Turner et al. suggest that, in addition to other previously identified information such as activity, location, or calendar data, the sender of the notification could prove to be an important factor to estimate the receptivity of a user to a notification.

5.2.3 Availability in homes

Inferring the availability of home occupants for communication is a problem that combines several aspects of availability estimation in professional environments and on smart phones. Much like professional environments, the environment is relatively fixed and can be equipped with ambient sensors (as we have seen in previous chapters) that can help capture more information about the availability of occupants. The smart phone of a home occupant can itself be used in addition to these ambient sensors to provide more information related to availability.

The variability of situations in homes, like for smart phones, is quite a bit larger than in professional environments. Moreover, these situations and the corresponding collected data are highly dependent on the actual occupants themselves, which have particular habits and routines. In a professional environment, we can expect most users to exhibit shared habits due to their workplace's culture. As such, in professional environments, we can design availability inference approaches which rely on expert rules that are applied by users (such as indicating unavailability by closing their office door, or through their professional schedule). Such approaches do not seem well adapted to personal homes where each household has different routines, habits, and no particular constraints to change them.

We present 2 papers on availability estimation in homes in Section 5.2.3.1 and Section 5.2.3.2.

5.2.3.1 Estimating availability through audio-visual features

Takemae et al. propose in [143] an availability inference approach to help manage remote communication attempts in the smart home, much like our communication assistance service example. This approach relies on audio and visual data, on which they propose to extract a number of audio-visual features: voice power, frequency of changes in voice power, motion near a table area, changes in the location of occupants, etc.

Availability is then estimated using a support vector regression algorithm (based on the SVM). Their experiments highlight some correlations between truth values of availability and availability computed by their approach. These results suggest that audio and visual data provide some insight on the availability of occupants in homes.

However, as discussed in Chapter 2, recording audio and visual data is generally unacceptable in smart homes. Issues of privacy and acceptability created by these modalities of data collection mean that it is unlikely that a general public smart home system will contain such sensors. As such, we should ideally strive to not use these modalities for availability estimation.

5.2.3.2 Correlations between context dimensions and availability

Nagel et al. report in [108] a set of statistical studies on the link between availability for communication at home and other context dimensions. In particular, they study the influence of identity, place, activity, and companionship (e.g. the number of people with the target occupant) on availability for communication. Experimental results show that some occupants are more often available than others, that some places are highly correlated to availability while other places are not, and that some activities are useful to infer availability. Nagel et al. assume that the context dimension of time should be an important factor to infer availability, but their study does not highlight this assumption.

In a following paper by Nagel et al. [109], they introduce the notion that availability should not be shared identically depending on the potential correspondent. For example, an occupant might want to disclose their availability fully to their close family, while not sharing any availability information to simple acquaintances.

Through these 2 papers, Nagel et al. experimentally showed that context information can be used to infer availability in homes.

5.2.4 Discussion

We have seen in this section that the problem of availability estimation can be quite different depending on the application it is used in.

In professional environments, users respect a number of shared habits of their workplace. In addition, explicit availability indicators such as professional schedules are usually available. This makes it possible to infer availability using rule-based systems.

Such approaches cannot be easily applied when estimating availability on smart phones or in homes. Indeed, these two cases present much more variability in terms of possible situations, and personalization to specific users appears to become essential. Statistical methods that use training data thus may be better adapted in this cases.

In particular, we saw that Nagel et al. suggest to infer availability in homes from high level context dimensions such as place and activity. Takemae et al. used low-level features exctracted from sensor data instead, which we believe will be less useful for availability inference. Indeed, it is unclear that availability for communication will significantly influence atomic actions of occupants, in the same way that the activity they aimed to do will.

Many of the works we surveyed dealt with a single communication modality, such as e-mails or phone calls. Fogarty et al. proposed instead a system that actually suggests preferable communication modalities depending on the availability of occupants. Extending this idea, we believe that availability inference should take into account the modality used to initiate the communication. For example, an occupant will probably be more available to receive an e-mail than to receive a phone call when they are sleeping at night.

Similarly, Nagel et al. suggested that occupants of homes might not want to share their availability identically depending on the potential correspondent. Extending this idea, we believe that availability inference should take into account the correspondent that attempts to communication with the occupant. For example, an occupant will probably be more available for their close family than for a stranger.

Our contributions to the problem of availability estimation thus rely on modelling availability as a function of other context dimensions. We will extend the list of important context dimensions with correspondents and modalities, which were not directly considered by other works. We will reuse our activity

recognition and prediction contributions presented in Chapter 3 and Chapter 4 to obtain necessary information on availability.

5.3 Availability as a function of context

In this section, we present our contributions to the problem of availability inference. We begin by introducing our general context-based availability inference methodology in Section 5.3.1. Then, we discuss in more details about the 3 secondary context dimensions required by an availability inference system: correspondents (in Section 5.3.2), modalities of communication (in Section 5.3.3), and availability (in Section 5.3.4). Finally, we present in Section 5.3.6 a number of possible assumptions about our context-based availability inference approach. We will experimentally check the validity of these assumptions in order to support our availability inference general methodology.

5.3.1 Availability as a function of context

Primary context, as presented in Section 2.1.2.1, contains 4 context dimensions that are often *necessary* to provide context-aware services. However, primary context is not always *sufficient*. Other context dimensions are often required to provide such services. In [46], Dey and Abowd claim that all context dimensions that are not primary dimensions (which they call second level context) can be indexed by primary context. For example, the ambient temperature for an occupant can be found by querying the temperature sensor which is in the same place as the occupant; ambient temperature is thus in this case indexed by place.

Although this claim is reasonable for many context dimensions, it is possible to find examples of services which require non-primary context dimensions that *cannot* be indexed on identity, time, place or activity. One example of such a service is a communication assistant which manages incoming communications from outside. This service would advise outside correspondents on appropriate times at which to initiate communications, propose to use other communication modalities that would be preferable for the receiving occupant, or even automatically translate a communication from one modality to another (for example, turn an incoming call into an SMS using speech-to-text algorithms).

To ensure such functions, this communication assistant requires information about the primary context: the identity of the occupant that the outsider is trying to communicate with is obviously required; the time at which this communication is initiated may have an impact on the acceptability of that communication; the current location of the occupant has an impact on the available communication modalities; the activity of the occupant may impact the types of communication modalities that can be used (e.g. an occupant taking a shower cannot answer a phone call but can receive a mail). In addition to this primary context, the communication assistant also requires information about the *correspondent* which initiated the communication, about the communication *modality* initially used by

that correspondent, and about the *availability* of the occupant being called. The correspondent and the communication modality usually cannot be indexed on primary context dimensions since the communication is initiated from outside the home, and thus potentially independently of any event that is monitored by the smart home. The claim of Dey and Abowd that all non-primary context dimensions are indexable on primary context is thus not always true. In general, the context dimensions needed to provide a particular smart home service will be a superset of the primary context.

More formally, we thus define the availability of an occupant $\alpha \in \mathcal{A}v$ to be the image through a function Av

$$\text{Av} : \mathcal{I} \times \mathcal{T} \times \mathcal{P} \times \mathcal{A} \times \mathcal{C} \times \mathcal{M} \longrightarrow \mathcal{A}v, \tag{5.1}$$

of the corresponding context tuple (identity, time, place, activity, correspondent, modality) in $\mathcal{I} \times \mathcal{T} \times \mathcal{P} \times \mathcal{A} \times \mathcal{C} \times \mathcal{M}$.

5.3.2 Correspondents

In theory, any person initiating a communication can be considered to be part of the set of correspondent values. In practice however, the number of possible correspondents would be very high in such a case. The occupant would have to indicate their availability preferences for every person that has or may communicate with them, which in most cases will be too inconvenient.

As such, we propose to use categories of correspondents, so that each correspondent can be placed in one of these categories. The occupant will thus only have to indicate their availability for each of the category, instead of every single individual correspondent. We have identified the following categories of correspondents for which an occupant might choose significantly different availability preferences:

- Close relatives;
- Distant relatives;
- Professional colleagues;
- Professional supervisors;
- Friends;
- Acquaintances;
- Strangers.

These categories are fairly similar to those identified by Nagel et al. in [109].

In future smart home systems, allowing occupants to create their own correspondent categories might allow more accurate availability estimation. For example, an occupant might have very specific availability preferences for certain individuals who would need to be in their own category (e.g. their spouse).

5.3.3 Modalities of communication

Similarly to correspondents, listing each possible modality of communication will often result in a list too large for convenient availability labelling. In much the same way, we thus propose to use categories of modalities. In modalities, different hierarchies can be identified: first, we can differentiate *synchronous* modalities from *asynchronous* modalities; second, we can separate modalities in each of these 2 categories based on the physical medium used to communicate (voices, videos, or text); last, we can further divide these subcategories based on the type of devices used. Such subdivisions lead to the following hierarchical domain of values of modality:

- Synchronous
  - Voice
    - Landline phone
    - Mobile phone
    - Computer
  - Video
    - Mobile phone
    - Computer
  - Text
    - Mobile phone
    - Computer
- Asynchronous
  - Voice
    - Landline phone
    - Mobile phone
  - Text
    - Mobile phone
    - Computer

In practice, categories of modality will change over time, as technology and standard uses evolve. Nevertheless, we will use this hierarchy of categories in our thesis as we expect it to cover most communication modalities currently used.

5.3.4 Values of availability

There does not seem to be a consensus on what values should be used to represent the availability of a person. We find the following non-exhaustive list of domains of values in various papers:

- 3-point scale (“Low”, “Medium”, “High”) [78];
- 4-point scale (“Highly unavailable”, “Unavailable”, “Available”, “Highly unavailable”) [54];
- 5-point scale (from “Highly unavailable” to “Highly available”) [53, 41];
- 5-point relative scale (from “Least interruptible” to “Most interruptible”) [143];
- 4 qualitative values (“For a quick question”, “For a discussion”, “Soon”, “Not at all”) [105].

As far as we have seen, these domains of values are chosen with no particular justifications. We even find two different scales in [54] and [53], even though both works share authors, with seemingly no justification for this change. In a survey by Turner et al., the same observation is stated: there does not seem to be any consensus on which domain of values to use for availability; both increasing scales and qualitative sets are used, with no clear justifications [151].

Therefore, we decided to adopt the following scale of availability values:

-2: Definitely unavailable;

-1: Preferably unavailable;

0: No opinion, does no know;

1: Preferably available;

2: Definitely available.

This scale has intrinsic advantages both for occupants' convenience as well as algorithmic processing. Indeed, this scale contains both hard decisions ("Definitely unavailable", "Definitely available") and soft decisions ("Preferably unavailable", "Preferably available") on availability. This allows an occupant some flexibility depending on how available they feel. Moreover, the occupant can choose to not give an opinion on their availability. This is important from a user standpoint, as one's own availability is often difficult to evaluate. The symmetry of the scale simplifies this evaluation of availability: the occupant can compare their current situation to previous ones and assign opposite availability to opposite situations and similar availability to similar situations. From a computing standpoint, this scale is numeric and ordered. As such, regression and similar numerical techniques can be applied on availability data in a meaningful way.

5.3.5 Inferring availability

It is not clear which function Av, as defined in Section 5.3.1, will best model the relationships between primary (identity, time, place, activity, correspondent, modality) and the availability of an occupant. In our thesis, we initially propose the following simple inference approach, which we illustrate on Figure 5.1:

1. much like for activity recognition and prediction, we assume that there is only one occupant in the home, or that we have an identification system at hand;
2. we use our place-based activity recognition approach to obtain information about place p and activity a for the current situation;
3. we select a correspondent-modality couple (c, m);
4. we retrieve the availability values in the training set for all situations (p, a, c, m);
5. the inferred availability for the current situation is the average of all previously retrieved availabilities.

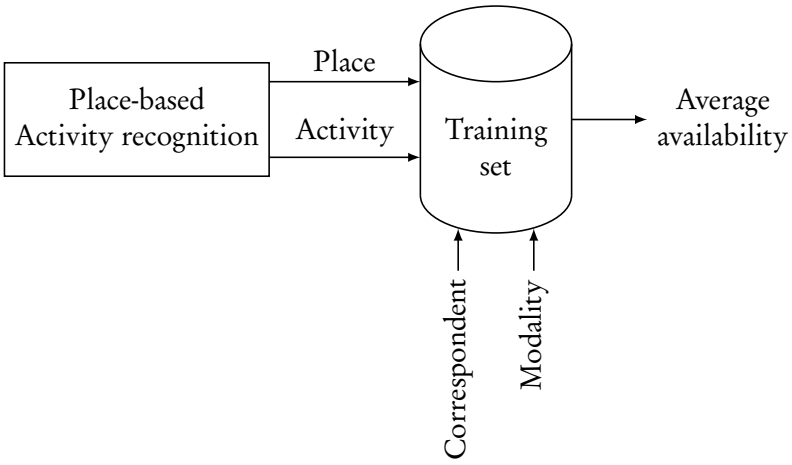

Figure 5.1 – Availability inference workflow.

Experimentally evaluating the performances of this averaging approach will allow use to get baseline results on availability inference and thus on the difficulty of this problem. Context information about time is not directly used in this inference function, although activities in particular can indirectly be related to time. Nevertheless, future work (not addressed in this thesis) should focus on introducing time information in the inference process, and on the design of a more complex availability inference function in general.

5.3.6 Labelling complexity under various assumptions

In our thesis, we chose to model availability as a function of various context dimensions. In particular, we conjectured that activities, correspondents, and modalities will greatly influence the availability of an occupant. However, introducing such dependencies between availability and these context dimensions means that more labelled input data is required to infer availability accurately. Moreover, it requires that an accurate activity recognition system is available (which, as we have seen in Chapter 3, is not a given).

Therefore, in order to validate the importance of introducing such dependence between availability and activities, correspondents, and modalities, we propose 4 simplifying assumptions one can make about our model. We will then experimentally study the performances of availability inference under each of these 4 assumptions in Section 5.4.4, compared to our initial approach under no assumption.

Let $(I, T, P, A) \in \mathcal{I} \times \mathcal{T} \times \mathcal{P} \times \mathcal{A}$ be a set of given identity, time, place, and activity of a situation. We can then construct the following 4 assumptions:

Assumption 5.1 (Modality independence)**.** *For a given correspondent $C \in \mathcal{C}$, the availability of an occupant is independent of the modality used, i.e.* $\forall (m_1, m_2) \in \mathcal{M}^2, \text{Av}(I, T, P, A, C, m_1) = \text{Av}(I, T, P, A, C, m_2)$.

Assumption 5.2 (Correspondent independence)**.** *For a given modality $M \in \mathcal{M}$, the availability of an occupant is independent of the correspondent initiating the communication, i.e.* $\forall (c_1, c_2) \in \mathcal{C}^2, \text{Av}(I, T, P, A, c_1, M) = \text{Av}(I, T, P, A, c_2, M)$.

Assumption 5.3 (Correspondent-and-Modality independence)**.** *The availability of an occupant is independent of the correspondent initiating the communication and independent of the modality used, i.e.* $\forall (c_1, c_2) \in \mathcal{C}^2, \forall (m_1, m_2) \in \mathcal{M}^2, \text{Av}(I, T, P, A, c_1, m_1) = \text{Av}(I, T, P, A, c_2, m_2)$.

Assumption 5.4 (Activity independence)**.** *For a given correspondent $C \in \mathcal{C}$ and modality $M \in \mathcal{M}$, the availability of an occupant is independent of the activity that they are performing, i.e.* $\forall (a_1, a_2) \in \mathcal{A}^2, \text{Av}(I, T, P, a_1, C, M) = \text{Av}(I, T, P, a_2, C, M)$.

Under each of these 4 assumptions, the initial averaging approach for availability inference changes slightly: the availability of an instance from activity class $a \in \mathcal{A}$, for a couple of correspondent and modality $(c, m) \in \mathcal{C} \times \mathcal{M}$, is the average (rounded to the closest integer) of

Assumption 5.1: all availabilities with correspondent c of instances of a in the training set;

Assumption 5.2: all availabilities with modality m of instances of a; in the training set;

Assumption 5.3: all availabilities for all couples, regardless of their correspondent and modality, of instances of a in the training set;

Assumption 5.4: all availabilities for the couple (c, m) of instances of activity classes that can occur in the same place as a in the training set.

Under each of these 4 assumptions, the number of availability values required by the system is different. For example, under the modality independence assumption, it is sufficient to label availability once for all modalities, if the activity and correspondent are fixed. In general, the number of availability values required is

No assumption: $|\mathcal{A}| \times |\mathcal{C}| \times |\mathcal{M}|$;

Assumption 5.1: $|\mathcal{A}| \times |\mathcal{C}|$;

Assumption 5.2: $|\mathcal{A}| \times |\mathcal{M}|$;

Assumption 5.3: $|\mathcal{A}|$;

Assumption 5.4: $|\mathcal{P}| \times |\mathcal{C}| \times |\mathcal{M}|$.

As such, if any of these assumptions is valid (i.e. they don't greatly decrease availability inference performances), the number of labels required for availability would highly decrease. This would greatly improve the acceptability of a communication assistance service based on availability inference, since labelling is one of the main constraints imposed on occupants by such a system. In particular, given the number of categories of correspondents (7) and modalities (11) we identified, and given an average number of 20 activity classes, there would be 1540 different combinations to label, which is too large.

5.4 Experiments

In the following section, we experimentally study the performances of our availability inference approach using the Orange4Home dataset. We present the availability labelling process in Orange4Home in Section 5.4.1. In Section 5.4.2, we discuss which metrics to use to evaluate the performance of availability inference. In Section 5.4.3, we analyse the performances of our availability inference approach, following a place-based activity recognition approach. In Section 5.4.4, we examine the validity of various independence assumptions we established in Section 5.3.6. Finally, in Section 5.4.5, we study the impact of the activity recognition step on the performances of the availability inference step.

For reference, some of the results reported in this section were previously published in [37].

5.4.1 Availability for communication in Orange4Home

In order to evaluate the performances of our availability inference approach, we need a labelled datasets of activities and availability for communication in the home. As far as we know, no such dataset existed in the literature. Therefore, we chose to label information about availability during the Orange4Home data collection, in addition to the labelling of activities which we had presented in 2.3.2.

The categories of correspondents, modalities, and availability values used in Orange4Home are those presented in Section 5.3. Since the number of Correspondent-Modality couples is high (7 correspondent and 11 modalities = 77 couples), it would be quite cumbersome to label each possible couple with an availability value every time an activity instance is performed. Therefore, the occupant initially selected their preferred availabilities for each activity class before the data collection phase started. An example of such preset availabilities for activity "Watching TV" in the Living room is presented in Table 5.1.

The Occupant thus only had to marginally change their availability as they desired, before beginning an activity. Such predefined availabilities allowed the occupant of the Orange4Home dataset to label *in situ* their availability at the same time as activity, instead of labelling availability after the data collection phase. As such, their recorded availability should better represent their actual desired availability at the time they performed their activities. In addition, labelling all Correspondent-Modality couples for each activity instance, while being only cumbersome in a research project, is an unrealistic task to ask in a real general public smart home system, given the number of possible couples.

5.4.2 Evaluation metrics

In activity recognition or prediction, evaluating the correctness of a decision is usually straightforward: either the decision is correct (i.e. it corresponds to

	Synchronous							Asynchronous			
	Voice			Video		Text		Voice		Text	
Correspondent	L	M	C	M	C	M	C	L	M	M	C
Close Relatives	−1	1	−2	−2	−2	2	−2	0	0	2	2
Distant Relatives	−1	1	−2	−2	−2	2	−2	0	0	1	2
Prof. colleagues	−2	−2	−2	−2	−2	−2	−2	−2	−2	−2	2
Prof. supervisors	−2	−2	−2	−2	−2	−2	−2	−2	−2	−2	2
Friends	−1	1	−2	−2	−2	2	−2	0	0	1	2
Acquaintances	−1	1	−2	−2	−2	1	−2	0	0	−1	2
Strangers	−1	1	−2	−2	−2	1	−2	0	0	−1	2

Table 5.1 – Preset availabilities for activity "Watching TV" in the Living room in the Orange4Home dataset.

reality), or it is not. For availability inference, we need to take into account a number of subtleties.

The first of those subtleties lies in the gradation scale used to measure availability. In our thesis, we use an ordered scale of 5 grades, presented in Section 5.3.4, from "Definitely unavailable" (-2) to "Definitely available" (2). Inference errors on such an ordered scale are thus quite different from errors on discrete sets of classes, as in activity recognition and prediction where errors are all binary.

Indeed, in availability inference, some errors are worse than others. For example, mistaking "Definitely available for "Definitely unavailable" can lead to unacceptable situations where a communication assistant would interrupt the occupant inappropriately based on that incorrect availability estimation. On the other hand, mistaking "Definitely available" for "Preferably available" is a more acceptable error, as in both cases the occupant is available. Formally, we thus require that availability inference errors with large absolute values (e.g. 4 when mistaking "Definitely available" for "Definitely unavailable") should penalize the inference method more than errors with small absolute values (e.g. 1 when mistaking "Definitely available" for "Preferably available").

The Root Mean Square Error (RMSE) is a measure commonly used to evaluate regression models. RMSE attributes more weight to errors with large absolute values, due to its quadratic terms. Let $\mathcal{X}$ be the set of activity instances evaluated. We have

$$\text{RMSE} = \sqrt{\frac{1}{|\mathcal{X}|}\sum_{i\in\mathcal{X}}(\widehat{\text{Av}}_i - \text{Av}_i)^2}. \tag{5.2}$$

As such, RMSE seems to be a good choice to evaluate the performance of availability inference.

However, the second subtlety in measuring the performance of an availability inference approach lies on the duration of activities. Indeed, inferring an incorrect availability from an activity instance is much worse if that activity instance is

particularly long. For example, if the system incorrectly deduces that the occupant is "Definitely unavailable" during an activity that lasts 4 hours, when they in fact are "Definitely available", then a very long segment of time of availability is lost, which can lead to unnecessary communication delays between the occupant and their correspondents. If such a mistake occurs for a short activity, then it will have less impact on the quality of the communication assistance service provided to the occupant.

Such a relationship between errors and durations of activities is not captured by the standard RMSE. Therefore, we propose to use a Duration-Weighted Root Mean Square Error (DWRMSE):

$$\text{DWRMSE} = \sqrt{\sum_{i \in \mathcal{X}} w_i (\widehat{\text{Av}}_i - \text{Av}_i)^2}, \tag{5.3}$$

$$w_i = \frac{d_i}{\sum_{j \in \mathcal{X}} d_j},$$

where d_i is the duration of activity instance i. If $w_i = \frac{1}{|\mathcal{X}|}$ for all $i \in \mathcal{X}$, DWRMSE is strictly equivalent to RMSE. DWRMSE penalize large absolute errors as well as errors for long activity instances, which are both desirable for availability inference.

In addition, we also report the error rate in experiments, that is the number of inferred availabilities that are not equal to ground truth, divided by the total number of inferred availabilities (i.e. $1 -$ accuracy). The error will complement the DWRMSE: the former will indicate the number of availability inference errors, while the latter will indicate how bad these errors are.

5.4.3 Availability inference following activity recognition

We report in Table 5.2 the DWRMSE and error rates of our availability inference approach averaged for each category of correspondents and modalities. The necessary preliminary activity recognition step was fulfilled by an MLP place-based approach, which was the most accurate single-classifier method on Orange4Home with an F_1 score 93.05% (full results in Table 3.4) as reported in Section 3.4.1.2. The same training protocol is used: the first 2 weeks of data serve as a training set, the third week as a validation set, and the last week as the test set.

We can first observe that the average error rate is quite small (0.015), despite using a simple averaging Av function (as described in Section 5.3.5). This suggests that, for the most part, the occupant did not change their availability preferences a lot from one activity instance to the next on average.

However, we can see that not all categories of correspondents and modalities are equally well inferred. For example, "Professional colleagues" and "Professional supervisors" have significantly lower error rates (0.002 and 0.006 respectively) than "Close relatives" or "Distant relatives" (0.019 for both). A possible explanation is that availability for colleagues is almost always exclusively limited to

		DWRMSE	Error rate
Correspondent	Close relatives	0.126	0.019
	Distant relatives	0.119	0.019
	Prof. colleagues	0.222	0.002
	Prof. supervisors	0.147	0.006
	Friends	0.128	0.020
	Acquaintances	0.489	0.023
	Strangers	0.137	0.019
Modality	Sync. Voice landline	0.067	0.007
	Sync. voice mobile	0.133	0.007
	Sync. voice computer	0	0
	Sync. video mobile	0.220	0.003
	Sync. video computer	0.220	0.003
	Sync. text mobile	0	0
	Sync. text computer	0	0
	Async. voice landline	0.152	0.037
	Async. voice mobile	0.180	0.048
	Async. text mobile	0.521	0.044
	Async. text computer	0.380	0.022
	Average	0.232	0.015

Table 5.2 – DWRMSE and error rate of availability inference averaged by correspondent, by modality, and on average, based on an MLP place-based activity recognition step.

working hours and identical from one day to the next, which is not the case for relatives. Despite lower error rates, the DWRMSE for "Professional colleagues" and "Professional supervisors" is actually larger (0.222 and 0.147 respectively) than for "Close relatives" and "Distant relatives" (0.126 and 0.119 respectively). This indicates that, in the rare situations where availability for colleagues differs from the typical preferences of the occupant, the change is quite important: notably, we can conjecture that the occupant rarely becomes "Definitely available" in a non-working situation where they typically are "Definitely unavailable", and vice-versa for working situations. Such changes cannot be well-captured by a naïve averaging inference.

In general, we observe that "Close relatives", "Distant relatives", and "Friends" have the lowest DWRMSE. These categories are emotionally closer to the occupant than the other 4 categories of correspondents. As such, the occupant might not be as inclined to indicate that they are "Definitely unavailable" for such people, which limits the possibility of large inference errors.

Similarly, we see that asynchronous modalities have much worse error rates and DWRMSE compared to synchronous modalities on average. This can be

explained by the fact that synchronous modalities require, as their name suggests, direct and real-time interactions. This imposes restrictions on the availability of the occupant, depending on the environment. For example, it is physically impossible to answer a landline phone call, located in the living room on the ground floor, while showering in the bathroom on the first floor. Therefore, many availability values for such synchronous modalities are implicitly imposed by the activity performed, which will reflect in the preferences set by the occupant. For asynchronous modalities, no such real-time interaction constraint exists; the occupant can therefore change their availability more freely.

Such observations should, in theory, be learned by the availability inference system during the training phase. However, many such changes occur sporadically, and might not have been captured in the training data. Moreover, our simple averaging approach is probably too naïve to fully capture such subtle changes in availability patterns.

5.4.4 Dependence of availability to other context dimensions

In this section, we experimentally study the behaviour of our availability inference approach under each of the 4 independence assumptions presented in Section 5.3.6: Modality independence (Assumption 5.1), Correspondent independence (Assumption 5.2), Correspondent-and-Modality independence (Assumption 5.3), and Activity independence (Assumption 5.4).

Under each of these 4 assumptions, the number of availability values the occupant has to preset or choose from, in the Orange4Home dataset, is respectively

Assumption 5.1: 20 activities $\times$ 7 correspondents $= 140$;

Assumption 5.2: 20 activities $\times$ 11 modalities $= 220$;

Assumption 5.3: 20 activities $= 20$;

Assumption 5.4: 8 places $\times$ 7 correspondents $\times$ 11 modalities $= 616$.

These numbers are all orders of magnitude smaller than the initial 1540 possible values to choose from if none of those assumptions are made. Lowering the number of availability values to preset and choose from is important to improve the acceptability and usability of services that rely on this information. The goal of the following experiments is thus to see whether these assumptions can be made without degrading availability inference too greatly.

We report in Table 5.3 the DWRMSE and error rates of availability inference under each of these 4 assumptions, compared to ground truth given by the occupant under no such independence assumptions. We see that error rates and DWRMSE under each of these 4 assumptions are greatly degraded, compared to our first results when no assumption was made (0.015 error rate and 0.232 DWRMSE, as reported in Table 5.2). We can note in particular that the error rates under Assumption 5.2 (Correspondent independence) and Assumption 5.3 (Correspondent-and-Modality independence) are close to 50% (42.7% and 48.9% respectively), which is obviously unacceptable if we were to use this inferred

	DWRMSE	Error rate
Modality independence	1.226	0.268
Correspondent independence	1.002	0.427
Correspondent-and-modality independence	1.393	0.489
Activity independence	0.637	0.151

Table 5.3 – DWRMSE and error rate of availability inference under various independence assumptions.

availability to provide services. Therefore, we can conclude that Assumption 5.2 and Assumption 5.3 are not valid on Orange4Home.

The error rate under Assumption 5.1 (Modality independence) is significantly smaller (0.268) than the previous 2 assumptions. However, it remains more than 17 times greater than the initial error rate we obtained under no assumption 0.015. Moreover, despite reaching a significantly lower error rate, availability inference under the Modality independence assumption presents higher DWRMSE than under the Correspondent assumption (1.226 compared to 1.226). This behaviour can be explained by the fact that long activities often imply preferred modalities of communication. For example, in Orange4Home, "Computing" in the Office and "Napping" in the Bedroom are typically long activity classes. In both cases, these activities specifically impact the choice of preferred modalities: when working on their computer, the occupant prefers synchronous communications related to their work; when napping, the occupant refuses any disruptive synchronous communication modalities. Under the modality independence assumption, these specificities cannot be captured and will thus degrade the quality of service for the occupant. Therefore, we can conclude that Assumption 5.1 is not valid on Orange4Home.

Finally, we observe that both the error rate and DWRMSE are the smallest (0.151 and 0.637 respectively) under Assumption 5.4 (Activity independence), among the 4 independence assumptions. This suggests that activity is less important for availability inference than correspondents and modalities identification. Nevertheless, the error rate observed is still 10 times greater than our initial results 0.015; making this assumption thus greatly degrades performance. Moreover, we conjecture that information about place partially captures the relationship between activity and availability. Indeed, the set of possible activities is limited to the place in which the occupant is. As such, their availability can be in part deduced from their location. For example, if the occupant is in the Bathroom, their availability will be similar for most Correspondent-Modality couples regardless of the actual activity they are performing, due to physical constraints of the room and behavioural similarities between activities that can occur in that place. Therefore, we can conclude that Assumption 5.4, while possibly acceptable, lead to high degradation in availability inference performance on Orange4Home.

		DWRMSE	Error rate
Correspondent	Close relatives	0.087	0.007
	Distant relatives	0.081	0.007
	Prof. colleagues	0.222	0.003
	Prof. supervisors	0.147	0.007
	Friends	0.091	0.008
	Acquaintances	0.483	0.011
	Strangers	0.119	0.008
Modality	Sync. Voice landline	0.067	0.007
	Sync. voice mobile	0.133	0.007
	Sync. voice computer	0	0
	Sync. video mobile	0.220	0.003
	Sync. video computer	0.220	0.003
	Sync. text mobile	0	0
	Sync. text computer	0	0
	Async. voice landline	0.119	0.019
	Async. voice mobile	0.131	0.019
	Async. text mobile	0.497	0.013
	Async. text computer	0.372	0.009
	Average total	0.220	0.007

Table 5.4 – DWRMSE and error rate of availability inference using true labels of activity averaged by correspondent, by modality, and on average.

5.4.5 Impact of activity recognition on availability estimation

We report in Table 5.4 the DWRMSE and error rates of our availability inference approach, when true activity labels are used instead of an imperfect activity recognition step. We can see that the error rate (0.007) is approximately two times smaller than in our previous experiments with an activity recognition step (0.015 as reported in Table 5.2). On the other hand, the DWRMSE are quite close in both cases (0.220 compared to 0.232).

This observation suggests that the majority of availability inference errors caused by incorrect activity recognition occur for short activities, which will not penalize the DWRMSE much. Indeed, the 8 errors made by the MLP place-based classifier occurred on short activity classes: “Cleaning” in the Bathroom, “Using the sink” in the Bathroom (twice), “Leaving” in the Entrance (thrice), “Preparing” in the Kitchen, and “Cleaning” in the Living room. Each of these activity classes have short durations on the order of a few minutes. These misclassifications are thus visible on the error rate for availability inference, but don’t have much impact on the DWRMSE.

We observe the same relative disparities in error rates and DWRMSE between

different categories of correspondents and modalities that we had already noted in Section 5.4.3. These differences are thus mostly due to the occupant's will to change their preferences, rather than side-effects of incorrect activity recognition. Therefore, a more elaborate availability inference algorithm should try to take into account that certain categories of correspondents and modalities are intrinsically more difficult to infer availability from. This is not currently captured by our averaging algorithm.

5.5 Conclusions

In this chapter, we presented our contributions on the problem of availability inference from context data, which is necessary in order to provide a context-aware communication assistance service in the home. We discussed how such specific services may require more than just primary context data (in this case, availability, correspondents, and modality), and how we can use activity recognition and prediction, which were the subject of Chapter 3 and Chapter 4, to help infer availability.

Based on our experiments we conclude that availability is indeed dependent not only on primary context dimensions of place and activity, but also on secondary context dimensions of correspondent and modality. Assuming independence from any of these 4 context dimensions leads to drastic degradations in availability inference performance. We have seen that a simple averaging approach is sufficient to infer availability with very little error rates. However, this approach does not seem to be able to capture subtle changes in availability preferences of the occupant from available context data. As such, further work is required to design more elaborate inference techniques, or to introduce new data sources to improve inference.

In this chapter, we decided to include 7 categories of correspondents and 11 categories of modalities. Taking into account the varying number of activity classes and places in a home, this leads to a large number of possible availability preferences depending on the value of each context dimension. This number of possibilities that need availability labelling is prohibitively large for a general public smart home system. Further work is thus required to either reduce the number of categories of correspondents and modalities (e.g. by combining some together), or to improve the ease of interaction for labelling availability in a general public smart home system (e.g. by analysing the actions of occupants in real communication situations).

As discussed in Section 5.4.4, while activity is an important factor for correctly inferring availability, place is already a key element of information for availability inference. Activities that occur in the same place tend to have similar availability preferences. Therefore, we recommend that the activity recognition approach used for availability inference is biased such that when it misclassifies an activity instance, it tends to decide in favour of another activity class of the same place.

As we have seen in Section 3.4.1.2, our place-based activity recognition approach does have this bias.

We have seen in Section 5.4.5 that our place-based approach tends to mostly misclassify short activity instances. Such misclassifications have little impact on DWRMSE, which we argue is a good indicator of performance for availability inference. Indeed, if availability for long-duration activities were incorrectly inferred, a communication assistant would not provide the appropriate service for long monolithic segments of time, which is less acceptable than for short-lived activities. Therefore, we recommend to concentrate effort on improving activity recognition for activities with long durations, rather than activities with short durations. Further work is required to see if our place-based activity recognition approach possess such bias, or if long activities in the Orange4Home dataset were simpler to recognize for reasons other than the approach's intrinsic behaviour.

Observing the reactions of occupants to incoming communications in the home could constitute a basis for future availability inference systems and dataset collection campaigns. Indeed, a predictive model of availability could benefit from feedback generated based on whether the occupant actually answered the commmunication (and thus was really available) or not. Similar ideas have been proposed for example by Smith et al. in [140].

Further work is required to evaluate the performance of availability inference following an activity prediction step. In this chapter, we only experimentally studied the behaviour of availability inference on current activity recognition. Accurate availability prediction is indeed necessary to implement a communication assistant with anticipatory capabilities. In addition, the PSINES activity prediction model we proposed in Chapter 4 could be enhanced to include variable nodes modelling availability. We would thus obtain a more direct availability prediction model which might prove to be more accurate.

In this thesis, we assumed that the home only contained one occupant at a time. As such, we ignored the problem of estimating availability of an occupant when they are already interacting with another occupant of the home. Such direct interactions surely have a major impact on availability for communication.

CHAPTER 6

Conclusions and perspectives

This chapter concludes the thesis with a summary of our contributions and their impact on the research domains of activity recognition, activity prediction, availability estimation, and smart homes in general. We finally discuss the limitations of our work and suggest new perspectives to address these problems.

6.1 Contributions and impact

The overarching idea of this thesis, upon which each of our contributions lies, is that context dimensions in the home are all interrelated. As such, we can use some of the knowledge we have on certain context dimensions to facilitate the discovery of other context dimensions. We illustrate this idea through our first contribution: the place-based activity recognition approach, which we discuss in more details in Section 6.1.1. Our contribution to activity prediction, through the PSINES DBN which we discuss in Section 6.1.2, also relies on this idea. Finally, we show how additional context dimensions, not part of primary context, can also be inferred from other context dimensions as well, through our contribution on availability inference, discussed in Section 6.1.3.

6.1.1 Place-based activity recognition

We proposed the place-based structure for activity recognition, where *a priori* context information on the location of sensors and activities are used to help in the recognition of activities and location of occupants. We argued that the

modular structure of the place-based approach allows for more accurate activity recognition, since classifiers can learn simpler place-based models and can be chosen among a variety of state-of-the-art classifiers, depending on the home. We experimentally showed that the place-based approach is indeed more accurate on 2 different datasets. We also showed that the place-based approach requires significantly shorter computing times, especially in the supervised training phase.

Improving activity recognition accuracy is an essential step in the development and acceptance of general public context-aware smart home systems. Activity is indeed an essential part of primary context, on which many potential services could rely. Incorrect activity recognition can imply inadequate services and thus low user satisfaction.

The modularity of the place-based approach and its training speed allows more flexibility in the supervised training phase of models. Using smaller place-based models, we can retrain only parts of the recognition system at a time, much faster, thus reducing downtimes and avoiding periods where the system does not provide any service (compared to a situation where a global model has to be trained all at once). This is especially useful in smart home environments where users may change their routines over time, or may change their sensor installation, which require retraining the recognition model.

6.1.2 Predicting activities using PSINES

We proposed to extend the CRAFFT DBN of Nazerfard and Cook [111] for activity prediction through 3 subcontributions, resulting in the PSINES DBN. We argued that the introduction of aggregated sensor data in PSINES will improve its prediction performance. We proposed that introducing non-Markovianness in the relationships between past and future activity instances will also improve prediction performance. Finally, we introduced a node modelling the cognitive state of the occupant in PSINES, to model the relationship between the will of the occupant and the activity they perform. We experimentally showed, on 6 different datasets, that each of these 3 subcontributions do improve activity prediction compared to CRAFFT, and that PSINES which uses all 3 improvements reaches on average the best prediction performances.

These results first show that activity prediction approaches in smart homes should not discard sensor data, which can contain valuable information that context dimensions do not capture. These results also show that non-Markovianness is essential to properly model the routines in most homes, which will not be properly captured by state-of-the-art first-order Markovian approaches. Modelling the cognitive state of occupants seems theoretically important, but marginally improves prediction performances. This may be explained by the fact that such cognitive states cannot be measured by sensors currently, so that we used latent nodes in our model, which necessarily bring less information than observed data.

6.1.3 Availability as a function of context

In accordance with our basic principle of interdependence between context dimensions in the home, we proposed to model the availability of occupants in homes as a function of other context dimensions. In particular, we showed that place, activity, correspondents, and modalities greatly impact the availability of an occupant. We experimentally showed that availability inference can be quite accurate with simple averaging strategies, assuming activity and place recognition are accurate.

We demonstrated through this contribution that secondary context dimensions can be inferred from primary context dimensions and other secondary context dimensions. Therefore, we reinforce through this study the idea that accurate primary context recognition and prediction are essential to provide context-aware smart home services, which will almost always rely on these context dimensions. In particular, we thus justify the need for further efforts on improving activity recognition and prediction approaches.

6.1.4 Orange4Home: a dataset of activities and availabilities in the home

We constructed a new dataset of activities and availabilities for communication in the home, called Orange4Home. Our data collection protocol allowed for the recording of realistic and varied routines through a large number of diverse ambient smart home sensors. We aimed at creating a new dataset with rich sensor and context data, that is representative of potential smart home environments that may exist in the future. We freely share this dataset with the scientific community [5].

As far as we know, Orange4Home is the only dataset containing labelled availability for communication in an instrumented home. By sharing this dataset, we thus hope to spark more interest on the problem of availability estimation in smart homes.

6.2 Limitations and perspectives

Each of our contributions presents limitations, either in terms of assumptions made, or in terms of results obtained. We discuss these limitations in the following subsections and present related perspectives to address these issues in future works.

6.2.1 A priori knowledge on context dimensions

In our place-based activity recognition approach, we assume that the location of sensors and activity classes is known in advance, in order to allocate each of them to the different place models. In current smart home systems, these information must be given by its users (i.e. the occupants of the home), which is

problematic: it requires effort from the occupants, and these information might become outdated as time goes by, requiring updating efforts. As such, it would be desirable that these information were discovered automatically.

Promising strategies for the discovering the location of sensors and activity classes include unsupervised machine learning approaches. In particular, we believe that clustering methods may allow to find groups of sensors and classes that are meaningful (in the home topology sense). Indeed, we can hypothesize that sensors located in the same place will often activate in a correlated fashion, whereas sensors in different places will not. Similar correlations may appear for activity classes. A major research problem introduced by this approach is in the choice of the best similarity metric for clustering to find such place-based correlations.

6.2.2 Accuracy of prediction models

Prediction accuracies we obtained in our experiments on the CASAS datasets using PSINES, while significantly higher than CRAFFT, remain much too low to be used in reliable smart home systems. Moreover, we hypothesize that predicting multiple future steps of activities in a row will lead to even lower accuracies. Results can be acceptable on datasets with high regularities in the routines of occupants (such as Orange4Home).

While our contributions show that sensors, non-Markovianness, and cognitive states are helpful to predict future activities, significant leaps are still required to reach high prediction accuracies. Designing predictive models capable of taking these leaps remains an open problem currently.

Further work is also required to predict durations and start times of activities in addition to the future activity class. Investigating the problem of occurrence prediction (from which we can solve sequence prediction) is thus a desirable perspective.

6.2.3 Availability prediction

In this thesis, we did not study availability prediction. A simple first set of experiments can consist in predicting activities using PSINES, on which we then infer availability for communication. Similarly to activity recognition, performances of the prediction algorithm will probably greatly impact availability inference.

Future work could include the extension of PSINES to directly include availability, through the inclusion of new nodes modelling availability, correspondents, and modalities. A comparative study of this new approach compared to the one presented in the previous paragraph could shed some light on whether availability can be inferred from predicted situations, or if availability should be directly predicted.

6.2.4 Labelling issues in supervised techniques

Each of our contributions to activity recognition, activity prediction, and availability inference requires a substantial labelled dataset of activities (and availabilities in the last case). Collecting labelled data is a perpetual problem of supervised machine learning approaches. In some research domains, large quantities of examples already exist (e.g. computer vision), and in others, labelled datasets can be provided by experts (e.g. medical assistance). In smart home research, few representative datasets exist (due to the lack of instrumented homes), and labelled datasets are generally not collected over very long periods of time (due to the constraints of living in an experimental setting, and the cost of labelling data).

Further work is thus required to simplify the collection of representative, labelled, long-spanned datasets of activities in the home. Potential improvements include the use of more suitable modalities for *in situ* labelling, such as vocal assistants (which have become more popular recently). Labelling could be provided occasionally through vocal exchanges with the assistant, thus reducing annoyance for occupants. Interaction loops where occupants would validate the behaviour of the system, explicitly or implicitly (through their actions), could also be used to continuously improve the training dataset.

Another area of improvements related to labelled datasets, which we explored in the place-based recognition approach, consists in designing algorithms that require less labelled data. In particular, unsupervised or semi-supervised techniques may be prove to be valuable in smart home systems.

6.2.5 Multi-occupant scenarios

In this thesis, we always assumed that only one occupant was present in the home at once, and their identity was not taken into account for activity recognition, activity prediction, or availability inference. Obviously, this assumption does not hold in general households, which very often contain multiple occupants. Moreover, occupants are regularly in the same place of the home, and will interact with each other frequently (which will have an impact on their activity and availability).

While the place-based activity recognition approach can be adapted (by modifying the decision fusion step) to multi-occupant scenarios when occupants are in different places, it cannot process more general situations such as the ones described in the previous paragraph. The same can be said for our contributions on activity prediction and availability inference. New models are thus required to alleviate the need for this assumption.

We express our concerns on the possibility of addressing such multi-occupant scenarios using only ambient smart home sensors. It is indeed doubtful that such low-level measures can capture the complexity of interactions between occupants. Wearable sensors and audio-visual sensors might provide more information in

such cases, but have their own share of problems (as discussed in Chapter 2).

6.2.6 Concept drift in smart homes

In our thesis, we used supervised machine learning techniques to build recognition or prediction models from a training dataset. However, we did not consider the problem of concept drift [148], that is, the change in statistical behaviour of variables over time, which causes models to become less and less accurate. In smart homes, concept drift can be mainly caused by changes in routines of occupants (e.g. new hobbies), or by changes in the environment (e.g. a new sensor is installed in the home).

Concept drift in the home can occur both progressively or abruptly: for example, a sensor can be removed from the home occasionally, or the occupant can move into a completely different home; similarly, an occupant can find new hobbies during the year that don't impact their routines much, or they can invite their partner to live permanently in their home which has a great and unpredictable impact on the initial model.

Further work is thus required to address the problem of concept drift, taking into account the specificities of smart homes mentioned above. We suggest in particular that solutions which can address cyclic concept drift should be valuable in smart homes. Indeed, in addition to the previous problems, we hypothesize that occupants will have cyclic changes in routines during the year, caused for example by changes in seasons, periods of vacations, weekends, etc.

6.2.7 Acceptability of context-aware smart homes

Assuming that a smart home system always provides useful services at appropriate times, there would still be acceptability issues for such technologies currently. Indeed, occupants might no be comfortable with the idea that their home is able to infer private and potentially sensitive information about their context (identifying who is in the home, recognizing their activities, etc.). We also discussed in previous chapters that some data sources (such as cameras and microphones) are generally badly perceived by occupants. Remote processing of personal data in the cloud and security of information in general are also sensitive topics. We proposed approaches that limited these concerns by excluding intrusive sensor categories and by using personalized models that do not require high computing power (thus alleviating the need for remote processing), but all acceptability issues are not yet addressed.

Furthermore, legal issues can arise in such systems. In particular, new regulations on personal user data can impose constraints on accessible data or storage durations, and thus on context-aware services.

Bibliography

[1] CASAS datasets. http://casas.wsu.edu/datasets. 22, 103

[2] List of sensors in Orange4Home. https://amiqual4home.inria.fr/files/2017/06/sensors_localisation.txt. 29

[3] "Predict". https://www.merriam-webster.com/dictionary/predict. 82

[4] Amiqual4Home. https://amiqual4home.inria.fr. 26

[5] Orange4Home: a dataset of routine daily activities in an instrumented home. https://amiqual4home.inria.fr/orange4home. 26, 141

[6] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499. Morgan Kaufmann Publishers Inc., 1994. 85

[7] Umut Akdemir, Pavan Turaga, and Rama Chellappa. An ontology based approach for activity recognition from video. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 709–712. ACM, 2008. 38

[8] Fadi Al Machot, Ranasinghe Ranasinghe, Johanna Plattner, and Nour Jnoub. Human activity recognition based on real life scenarios. In *IEEE International Conference on Pervasive Computing and Communications (PerCom) Workshops*, 2018. 41

[9] Muhammad Raisul Alam, Mamun Bin Ibne Reaz, and M. A. Mohd Ali. SPEED: An inhabitant activity prediction algorithm for smart homes. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(4):985–990, 2012. 86

[10] Muhammad Raisul Alam, Mamun Bin Ibne Reaz, and Mohd Alauddin Mohd Ali. A review of smart homes—past, present, and future. *IEEE*

Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(6):1190–1203, 2012. 18

[11] Piotr Augustyniak and Grażyna Ślusarczyk. Graph-based representation of behavior in detection and prediction of daily living activities. *Computers in biology and medicine*, 95:261–270, 2018. 89

[12] Akin Avci, Stephan Bosch, Mihai Marin-Perianu, Raluca Marin-Perianu, and Paul Havinga. Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In *Architecture of computing systems (ARCS), 2010 23rd international conference on*, pages 1–10. VDE, 2010. 37

[13] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics. *Foundations of Artificial Intelligence*, 3:135–179, 2008. 39

[14] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In *International Conference on Pervasive Computing*, pages 1–17. Springer, 2004. 41

[15] Mary Bazire and Patrick Brézillon. Understanding context before using it. In *International and Interdisciplinary Conference on Modeling and Using Context*, pages 29–40. Springer, 2005. 8

[16] Irad Ben-Gal. Bayesian networks. *Encyclopedia of statistics in quality and reliability*, 2007. 59

[17] Asma Benmansour, Abdelhamid Bouchachia, and Mohammed Feham. Multioccupant activity recognition in pervasive smart home environments. *ACM Computing Surveys (CSUR)*, 48(3):34, 2016. 36

[18] Samuel Berlemont, Grégoire Lefebvre, Stefan Duffner, and Christophe Garcia. Siamese neural network based similarity metric for inertial gesture classification and rejection. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–6. IEEE, 2015. 72

[19] Samuel Berlemont, Grégoire Lefebvre, Stefan Duffner, and Christophe Garcia. Class-balanced siamese neural networks. *Neurocomputing*, 273:47–56, 2018. 42

[20] Amiya Bhattacharya and Sajal K. Das. Lezi-update: An information-theoretic framework for personal mobility tracking in pcs networks. *Wireless Networks*, 8(2-3):121–135, 2002. 86

[21] Christopher M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer, 2007. 57, 58, 59

[22] Dario Bonino and Fulvio Corno. Dogont-ontology modeling for intelligent domotic environments. In *International Semantic Web Conference*, pages 790–803. Springer, 2008. 38

[23] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. 57

[24] Bruno Bouchard, Sylvain Giroux, and Abdenour Bouzouane. A smart home agent for plan recognition of cognitively-impaired patients. *Journal of Computers*, 1(5):53–62, 2006. 38

[25] Remco R. Bouckaert. Bayesian network classifiers in weka for version 3-5-7. 2008. 59

[26] Oliver Brdiczka, James L. Crowley, and Patrick Reignier. Learning situation models in a smart home. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):56–63, 2009. 36, 44

[27] Berardina De Carolis, Stefano Ferilli, and Domenico Redavid. Incremental learning of daily routines as workflows in a smart home environment. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(4):20, 2015. 85

[28] Liming Chen, Jesse Hoey, Chris D. Nugent, Diane J. Cook, and Zhiwen Yu. Sensor-based activity recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):790–808, 2012. 37

[29] Liming Chen, Chris D. Nugent, and Hui Wang. A knowledge-driven approach to activity recognition in smart homes. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):961–974, 2012. 38

[30] Luke Chen, Chris D. Nugent, Maurice Mulvenna, Dewar Finlay, Xin Hong, and Michael Poland. A logical framework for behaviour reasoning and assistance in a smart home. *International Journal of Assistive Robotics and Mechatronics*, 9(4):20–34, 2008. 38

[31] Sungjoon Choi, Eunwoo Kim, and Songhwai Oh. Human behavior prediction for smart homes using deep learning. In *RO-MAN*, volume 2013, page 173, 2013. 89

[32] Diane J Cook, Aaron S Crandall, Brian L Thomas, and Narayanan C Krishnan. CASAS: A smart home in a box. *Computer*, 46(7):62–69, 2013. 6, 22

[33] Diane J. Cook, Narayanan C. Krishnan, and Parisa Rashidi. Activity discovery and activity recognition: A new partnership. *IEEE transactions on cybernetics*, 43(3):820–828, 2013. 36, 41, 50

[34] James L. Crowley and Joelle Coutaz. An ecological view of smart home technologies. In *European Conference on Ambient Intelligence*, pages 1–16. Springer, 2015. 1, 2, 11

[35] James L. Crowley, Joëlle Coutaz, Gaëtan Rey, and Patrick Reignier. Perceptual components for context aware computing. In *International conference on ubiquitous computing*, pages 117–134. Springer, 2002. 8

[36] Julien Cumin, Grégoire Lefebvre, Fano Ramparany, and James L. Crowley. Human activity recognition using place-based decision fusion in smart homes. In *International and Interdisciplinary Conference on Modeling and Using Context*, pages 137–150. Springer, 2017. 67, 75

[37] Julien Cumin, Grégoire Lefebvre, Fano Ramparany, and James L. Crowley. Inferring availability for communication in smart homes using context. In *IEEE International Conference on Pervasive Computing and Communications (PerCom) Workshops*, 2018. 73, 129

[38] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989. 57

[39] Sajal K. Das, Diane J. Cook, Amiya Battacharya, Edwin O. Heierman, and Tze-Yun Lin. The role of prediction algorithms in the mavhome smart home architecture. *IEEE Wireless Communications*, 9(6):77–84, 2002. 85

[40] Carl De Boor. *A practical guide to splines*, volume 27. Springer-Verlag New York, 1978. 53

[41] Edward S. De Guzman, Moushumi Sharmin, and Brian P. Bailey. Should i call now? understanding what context is considered when deciding whether to initiate remote communication via mobile devices. In *Proceedings of Graphics interface 2007*, pages 143–150. ACM, 2007. 125

[42] Christian Debes, Andreas Merentitis, Sergey Sukhanov, Maria Niessen, Nikolaos Frangiadakis, and Alexander Bauer. Monitoring activities of daily living in smart homes: Understanding human behavior. *IEEE Signal Processing Magazine*, 33(2):81–94, 2016. 19

[43] Arthur P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, pages 325–339, 1967. 62

[44] Murat Deviren, Khalid Daoudi, and Kamel Smaïli. Language modeling using dynamic bayesian networks. In *4th International Conference on Language Resources and Evaluation-LREC 2004*, 2004. 98

[45] Anind K. Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001. 10

[46] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. In *In HUC'99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, 1999. 9, 10, 12, 46, 123

[47] Didier Dubois, Petr Hájek, and Henri Prade. Knowledge-driven versus data-driven logics. *Journal of logic, Language and information*, 9(1):65–89, 2000. 37, 45, 46

[48] Didier Dubois and Henri Prade. Possibility theory. In *Computational complexity*, pages 2240–2252. Springer, 2012. 63

[49] Stefan Duffner, Samuel Berlemont, Grégoire Lefebvre, and Christophe Garcia. 3d gesture classification with convolutional neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 5432–5436. IEEE, 2014. 37, 43

[50] Maria R. Ebling. Can cognitive assistants disappear? *IEEE Pervasive Computing*, 15(3):4–6, 2016. 17

[51] Iram Fatima, Muhammad Fahim, Young-Koo Lee, and Sungyoung Lee. A unified framework for activity recognition-based behavior analysis and action prediction in smart homes. *Sensors*, 13(2):2682–2699, 2013. 87

[52] Mathieu Fauvel, Jocelyn Chanussot, and Jon Atli Benediktsson. *Decision fusion for hyperspectral classification*. John Wiley & Sons, New York, NY, USA, 2007. 63

[53] James Fogarty, Scott E. Hudson, Christopher G. Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara Kiesler, Johnny C. Lee, and Jie Yang. Predicting human interruptibility with sensors. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1):119–146, 2005. 125, 126

[54] James Fogarty, Jennifer Lai, and Jim Christensen. Presence versus availability: the design and evaluation of a context-aware communication client. *International Journal of Human-Computer Studies*, 61(3):299–317, 2004. 118, 125, 126

[55] Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, and Alessandro Saffiotti. Robot task planning using semantic maps. *Robotics and autonomous systems*, 56(11):955–966, 2008. 14

[56] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: theory and practice*. Elsevier, 2004. 14, 15

[57] Fausto Giunchiglia. Contextual reasoning. In *Epistemologia, Special Issue on I Linguaggi e le Macchine*, 1992. 9

[58] Fausto Giunchiglia, Enrico Bignotti, and Mattia Zeni. Personal context modelling and annotation. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*, pages 117–122. IEEE, 2017. 9, 12

[59] Karthik Gopalratnam and Diane J. Cook. Online sequential prediction via incremental parsing: The active leZi algorithm. *IEEE Intelligent Systems*, (1):52–58, 2007. 85

[60] Crina Grosan and Ajith Abraham. *Rule-Based Expert Systems*, pages 149–185. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 38

[61] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993. 38

[62] Mathieu Guillame-Bert and James L. Crowley. Learning temporal association rules on symbolic time sequences. In *Asian conference on machine learning*, pages 159–174, 2012. 83, 85

[63] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009. 66

[64] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In *ACM sigmod record*, volume 29, pages 1–12. ACM, 2000. 85

[65] Jianguo Hao, Abdenour Bouzouane, and Sébastien Gaboury. Complex behavioral pattern mining in non-intrusive sensor-based smart homes using an intelligent activity inference engine. *Journal of Reliable Intelligent Environments*, 3(2):99–116, 2017. 82

[66] Robert Hirschfeld, Pascal Costanza, and Oscar Marius Nierstrasz. Context-oriented programming. *Journal of Object Technology*, 7(3):125–151, 2008. 8

[67] Yu-Jin Hong, Ig-Jae Kim, Sang Chul Ahn, and Hyoung-Gon Kim. Activity recognition using wearable sensors for elder care. In *Second International Conference on Future Generation Communication and Networking*, volume 2, pages 302–305. IEEE, 2008. 17

[68] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991. 57

[69] Matthieu Hourbracq, Pierre-Henri Wuillemin, Christophe Gonzales, and Philippe Baumard. Learning and selection of dynamic bayesian networks for non-stationary processes in real time. In *30th International Florida AI Research Society Conference, FLAIRS-30*, 2017. 92

[70] Chien-Ming Huang and Bilge Mutlu. Learning-based modeling of multimodal behaviors for humanlike robots. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 57–64. ACM, 2014. 99

[71] Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3118–3125. IEEE, 2016. 89

[72] Vikramaditya R Jakkula and Diane J. Cook. Using temporal relations in smart environment data for activity prediction. In *Proceedings of the 24th International conference on machine learning*, pages 20–24, 2007. 83

[73] Victor Kaptelinin and Bonnie A. Nardi. *Acting with technology: Activity theory and interaction design*. 2006. 13, 14

[74] Henry Kautz, Oren Etzioni, Dieter Fox, and Dan Weld. Foundations of assisted cognition systems. 2003. 89

[75] Eunju Kim, Sumi Helal, and Diane J. Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing*, 9(1), 2010. 13

[76] Younggi Kim, Jihoon An, Minseok Lee, and Younghee Lee. An activity-embedding approach for next-activity prediction in a multi-user smart space. In *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017. 89

[77] Donald Ervin Knuth. *The art of computer programming*, volume 3. Pearson Education, 1997. 65

[78] Yasumasa Kobayashi, Takahiro Tanaka, Kazuaki Aoki, and Kinya Fujita. E-mail delivery mediation system based on user interruptibility. In *International Conference on Human-Computer Interaction*, pages 370–380. Springer, 2015. 118, 125

[79] Hema S. Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2016. 89

[80] Narayanan C Krishnan and Diane J. Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10:138–154, 2014. 18, 36, 41

[81] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011. 18

[82] Paula Lago, Fréderic Lang, Claudia Roncancio, Claudia Jiménez-Guarín, Radu Mateescu, and Nicolas Bonnefond. The ContextAct@A4H real-life dataset of daily-living activities. In *International and Interdisciplinary Conference on Modeling and Using Context*, pages 175–188. Springer, 2017. 26

[83] Paula Lago, Claudia Roncancio, Claudia Jiménez-Guarín, and Cyril Labbe. Representing and learning human behavior patterns with contextual variability. In *International Conference on Database and Expert Systems Applications*, pages 305–313. Springer, 2017. 82

[84] Oscar D. Lara and Miguel A. Labrador. A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209, 2013. 19, 37

[85] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. 43

[86] Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 1998. 57

[87] Jina Lee, Stacy Marsella, David Traum, Jonathan Gratch, and Brent Lance. The rickel gaze model: A window on the mind of a virtual human. In *International Workshop on Intelligent Virtual Agents*, pages 296–303. Springer, 2007. 99

[88] Seon-Woo Lee and Kenji Mase. Activity and location recognition using wearable sensors. *IEEE pervasive computing*, 1(3):24–32, 2002. 17

[89] Grégoire Lefebvre, Samuel Berlemont, Franck Mamalet, and Christophe Garcia. BLSTM-RNN based 3D gesture classification. In *International Conference on Artificial Neural Networks*, pages 381–388. Springer, 2013. 43

[90] Adrien Marie Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot, 1805. 41

[91] Alekseĭ Nikolaevich Leont'ev. Activity, consciousness, and personality. 1978. 14

[92] Kang Li and Yun Fu. Prediction of human activity by discovering temporal sequence patterns. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1644–1657, 2014. 82, 85

[93] Beth Logan, Jennifer Healey, Matthai Philipose, Emmanuel Munguia Tapia, and Stephen Intille. A long-term evaluation of sensing modalities for activity recognition. In *International conference on Ubiquitous computing*, pages 483–500. Springer, 2007. 18

[94] Sawsan Mahmoud, Ahmad Lotfi, and Caroline Langensiepen. Behavioural pattern identification and prediction in intelligent environments. *Applied Soft Computing*, 13(4):1813–1822, 2013. 89

[95] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery*, 1(3):259–289, 1997. 85

[96] M Marufuzzaman, MBI Reaz, MAM Ali, and LF Rahman. A time series based sequence prediction algorithm to detect activities of daily living in smart home. *Methods of information in medicine*, 54(03):262–270, 2015. 85

[97] Sven Mayer, Lars Lischke, Pawel W. Wozniak, and Niels Henze. Evaluating the disruptiveness of mobile interactions: A mixed-method approach. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 406. ACM, 2018. 119

[98] John McCarthy. *Programs with common sense*. 1958. 38

[99] Georgios Meditskos, Efstratios Kontopoulos, and Ioannis Kompatsiaris. Knowledge-driven activity recognition and segmentation using context connections. In *International Semantic Web Conference*, pages 260–275. Springer, 2014. 38

[100] Alaeddine Mihoub. *Apprentissage statistique de modèles de comportement multimodal pour les agents conversationnels interactifs*. PhD thesis, Université Grenoble Alpes, 2015. 92

[101] Alaeddine Mihoub and Grégoire Lefebvre. Wearables and social signal processing for smarter public presentations. *Transactions on Interactive Intelligent Systems*, 2018. 99, 109, 110

[102] Bryan Minor, Janardhan Rao Doppa, and Diane J. Cook. Data-driven activity prediction: Algorithms, evaluation methodology, and applications. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 805–814. ACM, 2015. 82

[103] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2):90–126, 2006. 14

[104] Dorothy N Monekosso and Paolo Remagnino. Anomalous behavior detection: Supporting independent living. 2009. 89

[105] Martin Muhlenbrock, Oliver Brdiczka, Dave Snowdon, and J. L. Meunier. Learning to detect user activity and availability from a variety of sensor data. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, pages 13–22. IEEE, 2004. 125

[106] Kevin P. Murphy. The Bayes net toolbox for MATLAB. In *Computing Science and Statistics*, 2001. 103

[107] Kevin Patrick Murphy. Dynamic bayesian networks: representation, inference and learning. 2002. 92

[108] Kristine S. Nagel, James M. Hudson, and Gregory D. Abowd. Predictors of availability in home life context-mediated communication. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 497–506. ACM, 2004. 121

[109] Kristine S. Nagel, Ja-Young Sung, and Gregory D. Abowd. Designing home availability services. *Personal and Ubiquitous Computing*, 11(5):361–372, 2007. 122, 124

[110] Ehsan Nazerfard. Temporal features and relations discovery of activities from sensor data. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–16, 2018. 85

[111] Ehsan Nazerfard and Diane J. Cook. CRAFFT: an activity prediction model based on bayesian networks. *Journal of ambient intelligence and humanized computing*, 6(2):193–205, 2015. xx, xxii, 83, 90, 91, 93, 94, 101, 102, 103, 104, 105, 106, 140

[112] George Okeyo, Liming Chen, and Hui Wang. Combining ontological and temporal formalisms for composite activity modelling and recognition in smart homes. *Future Generation Computer Systems*, 39:29–43, 2014. 39

[113] George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. Ontology-enabled activity learning and model evolution in smart homes. In *International Conference on Ubiquitous Intelligence and Computing*, pages 67–82. Springer, 2010. 38

[114] George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing*, 10:155–172, 2014. 41

[115] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016. 43

[116] Kazushige Ouchi and Miwako Doi. A real-time living activity recognition system using off-the-shelf sensors on a mobile phone. In *International and Interdisciplinary Conference on Modeling and Using Context*, pages 226–232. Springer, 2011. 41

[117] Julien Pansiot, Danail Stoyanov, Douglas McIlwraith, Benny Lo, and G. Yang. Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems. In *4th international workshop on wearable and implantable body sensor networks*, pages 208–212. Springer, 2007. 18

[118] Han Saem Park and Sung Bae Cho. Predicting user activities in the sequence of mobile context for ambient intelligence environment using dynamic bayesian network. In *2nd International Conference on Agents and Artificial Intelligence, ICAART 2010*, 2010. 89

[119] Juha Parkka, Miikka Ermes, Panu Korpipaa, Jani Mantyjarvi, Johannes Peltola, and Ilkka Korhonen. Activity classification using realistic data from wearable sensors. *IEEE Transactions on information technology in biomedicine*, 10(1):119–128, 2006. 17

[120] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2847–2854. IEEE, 2012. 17

[121] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010. 14

[122] Davy Preuveneers, Jan Van den Bergh, Dennis Wagelaar, Andy Georges, Peter Rigole, Tim Clerckx, Yolande Berbers, Karin Coninx, Viviane Jonckers, and Koen De Bosschere. Towards an extensible context ontology for ambient intelligence. In *European Symposium on Ambient Intelligence*, pages 148–159. Springer, 2004. 38

[123] Joseph Rafferty, Chris D. Nugent, Jun Liu, and Liming Chen. From activity recognition to intention recognition for assisted living within smart homes. *IEEE Transactions on Human-Machine Systems*, 47(3):368–379, 2017. 38

[124] Fano Ramparany. Semantic approach to smart home data aggregation multi-sensor data processing for smart environments. *Sensors & Transducers*, 199(4):20, 2016. 40

[125] Fano Ramparany, Ravi Mondi, and Yves Demazeau. A semantic approach for managing trust and uncertainty in distributed systems environments. In *21st International Conference on Engineering of Complex Computer Systems (ICECCS)*, pages 63–70. IEEE, 2016. 40

[126] Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3):579–590, 2013. 17

[127] John P. Robinson, Philip E. Converse, and Alexander Szalai. *The use of time: Daily activities of urban and suburban populations in twelve countries*. 1972. 23

[128] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl,

Alois Ferscha, et al. Collecting complex activity datasets in highly rich networked sensor environments. In *Seventh International Conference on Networked Sensing Systems (INSS)*, pages 233–240. IEEE, 2010. 5, 13, 20

[129] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985. 57

[130] Michael S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1036–1043. IEEE, 2011. 82

[131] Alireza Sahami Shirazi, Niels Henze, Tilman Dingler, Martin Pielot, Dominik Weber, and Albrecht Schmidt. Large-scale assessment of mobile notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3055–3064. ACM, 2014. 119

[132] Bill N. Schilit and Marvin M. Theimer. Disseminating active map information to mobile hosts. *IEEE network*, 8(5):22–32, 1994. 8

[133] Albrecht Schmidt and Kristof Van Laerhoven. How to build smart appliances? *IEEE Personal Communications*, 8(4):66–71, 2001. 17

[134] Mohamed A. Sehili, Dan Istrate, Bernadette Dorizzi, and Jerome Boudy. Daily sound recognition using a combination of gmm and svm for home automation. In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1673–1677. IEEE, 2012. 17

[135] Pavel Senin. Dynamic time warping algorithm review. 2008. 59, 60

[136] Glenn Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976. 62

[137] Claude Elwood Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:623–656, 1948. 85

[138] Shailendra Singh, Abdulsalam Yassine, and Shervin Shirmohammadi. Incremental mining of frequent power consumption patterns from smart meters big data. In *Electrical Power and Energy Conference (EPEC), 2016 IEEE*, pages 1–6. IEEE, 2016. 88

[139] Philippe Smets. The combination of evidence in the transferable belief model. *IEEE Transactions on pattern analysis and machine intelligence*, 12(5):447–458, 1990. 63

[140] Jeremiah Smith and Naranker Dulay. Ringlearn: Long-term mitigation of disruptive smartphone interruptions. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pages 27–35. IEEE, 2014. 137

[141] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 17

[142] Timo Sztyler, Gabriele Civitarese, and Heiner Stuckenschmidt. Modeling and reasoning with ProbLog: an application in recognizing complex activities. In *IEEE International Conference on Pervasive Computing and Communications (PerCom) Workshops*, 2018. 43

[143] Yoshinao Takemae, Takehiko Ohno, Ikuo Yoda, and Shinji Ozawa. Estimating human interruptibility in the home for remote communication. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, pages 1397–1402. ACM, 2006. 121, 125

[144] Takahiro Tanaka, Ryosuke Abe, Kazuaki Aoki, and Kinya Fujita. Interruptibility estimation based on head motion and PC operation. *International Journal of Human-Computer Interaction*, 31(3):167–179, 2015. 117

[145] Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *Proceedings of the Second International Conference on Pervasive Computing, Vienna, Austria*, pages 158–175, 2004. 18, 23

[146] Daphne Townsend, Frank Knoefel, and Rafik Goubran. Privacy versus autonomy: a tradeoff model for smart home monitoring technologies. In *Annual International Conference of the IEEE on Engineering in Medicine and Biology Society*, pages 4749–4752. IEEE, 2011. 19

[147] Ming-Je Tsai, Chao-Lin Wu, Sipun Kumar Pradhan, Yifei Xie, Ting-Ying Li, Li-Chen Fu, and Yi-Chong Zeng. Context-aware activity prediction using human behavior pattern in real smart home environments. In *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*, pages 168–173. IEEE, 2016. 85

[148] Alexey Tsymbal. The problem of concept drift: definitions and related work. *Technical Report, Department of Computer Science, Trinity College Dublin*, 2004. 144

[149] Pavan Turaga, Rama Chellappa, Venkatramana S. Subrahmanian, and Octavian Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video technology*, 18(11):1473–1488, 2008. 37

[150] Alan M. Turing. Computing machinery and intelligence. *Mind*, LIX(236):433–460, 1950. 41

[151] Liam D. Turner, Stuart M. Allen, and Roger M. Whitaker. Interruptibility prediction for ubiquitous systems: conventions and new directions from a growing field. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pages 801–812. ACM, 2015. 117, 126

[152] Liam D. Turner, Stuart M. Allen, and Roger M. Whitaker. Reachable but not receptive: Enhancing smartphone interruptibility prediction by modelling the extent of user engagement with notifications. *Pervasive and Mobile Computing*, 40:480–494, 2017. 120

[153] Tim van Kasteren, Gwenn Englebienne, and Ben J. A. Kröse. Transferring knowledge of activity recognition across sensor networks. In *Proceeding of the Eighth International Conference on Pervasive Computing, Helsinki, Finland*, pages 283–300, 2010. 23

[154] Saeed V. Vaseghi. *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008. 55

[155] Sarvesh Vishwakarma and Anupam Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, 29(10):983–1009, 2013. 37

[156] Jie Wan, Michael J O'Grady, and Gregory MP O'Hare. Dynamic sensor event segmentation for real-time activity recognition in a smart home context. *Personal and Ubiquitous Computing*, 19(2):287–301, 2015. 17

[157] Jia-Ching Wang, Hsiao-Ping Lee, Jhing-Fa Wang, and Cai-Bei Lin. Robust environmental sound recognition for home automation. *IEEE Transactions on Automation Science and Engineering*, 5(1):25–31, 2008. 17

[158] Xiao Hang Wang, D. Qing Zhang, Tao Gu, and Hung Keng Pung. Ontology based context modeling and reasoning using owl. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 18–22. Ieee, 2004. 38

[159] David H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992. 62

[160] Chen Wu, Amir Hossein Khalili, and Hamid Aghajan. Multiview activity recognition in smart homes with spatio-temporal features. In *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras*, pages 142–149. ACM, 2010. 45, 46, 47, 50

[161] Shaoen Wu, Jacob B. Rendall, Matthew J. Smith, Shangyu Zhu, Junhong Xu, Honggang Wang, Qing Yang, and Pinle Qin. Survey on prediction algorithms in smart homes. *IEEE Internet of Things Journal*, 4(3):636–644, 2017. 85

[162] Naoharu Yamada, Kenji Sakamoto, Goro Kunito, Yoshinori Isoda, Kenichi Yamazaki, and Satoshi Tanaka. Applying ontology and probabilistic model to human activity recognition from surrounding things. *IPSJ Digital Courier*, 3:506–517, 2007. 38

[163] Abdulsalam Yassine, Shailendra Singh, and Atif Alamri. Mining human activity patterns from smart home big data for health care applications. *IEEE Access*, 5:13131–13141, 2017. 88

[164] Jihang Ye, Zhe Zhu, and Hong Cheng. What's your next move: User activity prediction in location-based social networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 171–179. SIAM, 2013. 83, 89

[165] Juan Ye, Graeme Stevenson, and Simon Dobson. KCAR: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing*, 19:47–70, 2015. 39, 46

[166] G. Michael Youngblood and Diane J. Cook. Data mining for hierarchical model creation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(4):561–572, 2007. 24

[167] Lotfi A. Zadeh. Fuzzy sets. In *Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh*, pages 394–432. World Scientific, 1996. 63

[168] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 283–286. AAAI Press, 1997. 85

[169] Chenyang Zhang and Yingli Tian. RGB-D camera-based daily living activity recognition. *Journal of Computer Vision and Image Processing*, 2(4):12, 2012. 17

[170] Shugang Zhang, Zhiqiang Wei, Jie Nie, Lei Huang, Shuang Wang, and Zhen Li. A review on human activity recognition using vision-based method. *Journal of healthcare engineering*, 2017, 2017. 37