UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI INGEGNERIA

Corso di Laurea Magistrale a Ciclo Unico
Ingegneria Edile-Architettura

Sviluppo di modelli Digital Twin per l'Ambient Assisted Living

Development of Digital Twin models supporting Ambient Assisted Living

a cura di:
Leonardi Binni

Relatore:
Prof. Berardo Naticchia

Correlatori:
Ing. Massimo Vaccarini
Ing. Francesco Spegni

A.A. 2020/2021

To those who have believed in me at least once

TABLE OF CONTENTS

This page intentionally left blank

ABSTRACT

L'invecchiamento della popolazione rappresenta una delle più grandi problematiche presenti all'epoca attuale su scala globale. Le stime delle Nazioni Unite lo definiscono un trend nettamente in crescita, destinato a non arrestarsi. Conseguentemente, anche il numero di persone che soffre di disordini cognitivi e delle cosiddette malattie geriatriche è in aumento. Tali problematiche non rendono la persona anziana in grado di completare, o persino intraprendere, azioni quotidiane nell'ambiente casalingo. La perdita dell'autonomia, soprattutto per le persone che vivono sole, rappresenta un grave deficit che le porta necessariamente a dover ricevere aiuto da familiari e/o assistenti. Nel caso in cui quest'ultimi non possano essere presenti, gli anziani sono spesso destinati ad essere accolti in specifiche case di cura. Per loro, perdere la propria autonomia significa non solo sentirsi un peso nei confronti di altre persone, ma anzitutto diventare consapevoli che le proprie incapacità, causate dalla malattia, renderanno la situazione irreversibile. L'Ambient Assisted Living ha come principale obiettivo proprio quello di supportare le persone anziane per un invecchiamento autonomo, sano ed attivo.

Parallelamente, il crescente interesse riguardante l'innovazione digitale nel mondo delle costruzioni e dell'ambiente costruito ha portato alla definizione dei Digital Twins e allo sviluppo di modelli pionieristici. Un Digital Twin è definito come una copia virtuale di un asset reale, rispecchiandone le caratteristiche fisiche, costruttive e comportamentali. Concetto nato per l'industria manifatturiera, è diventato oggi un tema di attualità e avanguardia anche nel mondo delle costruzioni. La ricerca in questo campo ha ad oggi sviluppato ambiziose linee guida per la definizione di Digital Twin su scala nazionale. A livello di edificio la ricerca lavora su più fronti: dalla formalizzazione di modelli che seguono le fasi di costruzione per il controllo dell'esecuzione e la verifica del rispetto delle condizioni di sicurezza a gemelli digitali volti all'ottimizzazione delle fasi di gestione del bene, in grado di valutare anche l'impatto della presenza umana negli edifici.

A tale riguardo, sono proposti in questa ricerca modelli Digital Twin a supporto dell'Ambient Assisted Living che costituiscono un layer "intelligente" e permettono la definizione di un Edificio Cognitivo. Nello specifico, il nostro prototipo di edificio

cognitivo ha come obiettivo il sostegno di persone anziane che soffrono di disordini cognitivi e vivono autonomamente. Il sistema presentato, basato su un'architettura a cinque strati, riconosce eventuali anomalie che potrebbero presentarsi all'interno dell'ambiente casalingo. In caso di presenza di anomalie, il sistema prende decisioni in autonomo a supporto dell'utente. Le anomalie possono essere causate da attività non eseguite correttamente, combinazioni errate delle stesse, confusione temporale, azioni pericolose e così via. Il loro riconoscimento deriva da un processo di sviluppo della conoscenza: dalla raccolta di dati grezzi tramite sensori visuali e non, passando attraverso un modello di conoscenza di basso livello per il riconoscimento delle azioni, fino ad arrivare ad un modello di conoscenza di alto livello capace di prendere decisioni in modo autonomo una volta individuate anomalie nello scenario. Ogni componente del sistema è stata sviluppata, testata e valutata relativamente a quattro funzioni principali: rappresentazione 3D in tempo reale dell'ambiente e dell'utente, riconoscimento delle attività perseguite dall'utente, ragionamento di alto livello su possibili anomalie per prendere decisioni in autonomia, interazione bidirezionale tra il sistema e l'utente.

I risultati delle sperimentazioni sulle varie funzioni del sistema evidenziano la loro efficacia. La rappresentazione 3D in tempo reale dell'ambiente e dell'utente risulta altamente affidabile attraverso l'utilizzo di algoritmi per il filtraggio dei dati, la stabilizzazione dell'avatar e la correzione della sua posizione nello spazio. Questo permette di avere dati in uscita verso le reti neurali nettamente più coerenti rispetto al semplice utilizzo del solo modello di tracciamento dello scheletro umano. Il modello di reti neurali impiegato per il riconoscimento delle attività eseguite dall'utente conferma la sua accuratezza, nonostante mostri una notevole complicatezza. Le reti Bayesiane si dimostrano uno strumento appropriato per la gestione di situazioni complesse e difficili da prevedere, che possono generarsi all'interno di uno scenario come quello preso in considerazione per questo lavoro. La semantizzazione contestualizzata di situazioni, scenari ed anomalie permette di creare modelli Bayesiani capaci di supportare il sistema nel prendere decisioni in completa autonomia, in base alle informazioni provenienti dal contesto stesso. Inoltre, i moduli Object-Oriented formalizzati in questo lavoro, non solo sono flessibili, ripetibili e scalabili, ma potrebbero essere usati anche per valutazioni di tipo abduttivo (diagnostica a posteriori). L'interazione sistema/utente proposta, sebbene riguardante una singola

situazione, serve a validare un modello bidirezionale (STT-TTS) che può essere esteso in fasi successive.
Questa ricerca, oltre che a sviluppare modelli Digital Twin per edifici cognitivi concretamente applicabili e flessibili, punta l'attenzione sulla necessaria innovazione digitale del mondo delle costruzioni, processo già in itinere ma con ancora grandi potenzialità da sfruttare, dunque sull'esigenza di costruire o ammodernare edifici non solo pensando alle componenti Hardware, definite dagli elementi tangibili, ma anche alla componente Software, definita da un layer cognitivo capace di rispondere alle esigenze degli utenti.

CHAPTER I

INTRODUCTION

Population ageing is one of the four “mega-trends” that currently characterize the mankind and will continue to have significant and lasting impacts on sustainable developments in the decades to come. UN estimated the number of people aged 65 years or over were slightly more than 700 million in 2019, and this number is sharply rising. It is indeed projected to double by 2050, reaching 1.5 million people. Whereas the global percentage of over 65-year-old people was at just 6% in 1990, it experienced a growth by 3% within 2019 (9%) and is going to rise further to 16% by 2050.

The impact of the ageing population has direct consequences on the elderly lifestyles. UN has indeed estimated that there will be a higher percentage of independent older people in the future. Secondly, the increase in the number of people aged over 65 directly affects the share of the population suffering from cognitive disorders. When consciousness and thought processes are disrupted (including problem solving, attention, perception, and language), patients are said to have cognitive disorders. People with these disorders may be disoriented, inattentive, have memory and language disturbances, and sensory irregularities including hallucinations. Those are consequently not able to complete, or even undertake, Activities of Daily Living (ADLs). Loss of autonomy, above all among older people living alone, is a serious deficit that necessarily leads them to receive assistance from third parties such as family members and/or caregivers. If the latter cannot be present, the elderly are often forced to be hosted in specific nursing homes. For these people, losing their autonomy would mean not only becoming a burden to other people, but first of all realizing that their own inabilities, caused by the disease, will turn the situation into an irreversible one.

Ambient Assisted Living (AAL) is focused on the enhancement of the living standards of the aged people through various assistive solutions. Exploiting state-of-the-art technologies, AAL develops either products or technological services as solutions to foster the environments inhabited by older people. In this regard, AAL is working on the improvement of the elderly’s autonomy in their ADLs to make feasible an autonomous ageing. Conceptual models for AAL application lie on globally shared policies that aim at supporting the elderly's activities within the home environment through tailored services able to foster the quality of their lives:

- *Strengthening social/informal connectivity*
- *Strengthening connectivity with health system*
- *Context Awareness*

The attention from international organizations on the theme of Ambient Assisted Living demonstrates the significant impact that it might have on the lives of the older people. In the last years, the progress in technology has allowed the development of new AAL applications. Thus, some envision in this domain has become reality. Above all, Artificial Intelligence systems have enabled significant improvements. The applications of AAL services are included in four main areas:

- Daily Task Facilitation
- Mobility Assistance
- Healthcare and rehabilitation
- Communication and social inclusion

The growing interest in digital innovation in the world of construction and built environment has led to the definition of Digital Twins (DTs) and the development of pioneering models. A Digital Twin is defined as a virtual copy of a real asset, mirroring its physical, constructive, and behavioural features. A concept originated in manufacturing, it has now become a cutting-edge topic in the construction industry and the built environment. What distinguishes a DT from any other digital model is its connection to the physical twin and its interoperability. Built upon data from the real asset, a digital twin releases value mainly by supporting enhanced decision making, which creates the opportunity for positive feedback into the physical twin. The advance of the *Industry 4.0* is making cyber-physical systems in the built environment practical, useful and affordable. The concept of DTs lies on data-driven design frameworks. One of the enabling technologies is the Internet of Things (IoT). Big Data fed Artificial Intelligence (AI) models and analytics systems. Machine Learning (ML) and Deep Learning (DL) systems are usually exploited for monitoring, predictions, simulations and feedback. DTs can operate at different scales and can adopt different approaches to modelling. In regard to the building's functionality, DT is a tool that can enhance efficiency, prevention and prediction of likely events during the whole building's life cycle. Nowadays, buildings are increasingly becoming equipped with IoT devices. A fully automatized building is best known as *Smart Building*. Digital Twin at a building level is a concept that goes beyond common smart buildings, defining cognitive and responsive environments. It describes the state of a system and its evolution over time, taking into account users as well.

The need to find valuable solutions for the support of the elderly population for an active and healthy autonomous ageing, and the theme of digital innovation in the field of construction and the built environment, encounter each other and define the foundations of this work of thesis. Digital Twins in Ambient Assisted Living define *Cognitive Buildings.* Those can learn at scale, reason with purpose and co-operate with users in a natural way. They learn and reason from the interactions with both users and environments where they are deployed, evolving with them and running *"What if?"* scenarios for predicting anomalies, behaviours and events.

The objective of this research is therefore the development of Digital Twin models supporting Ambient Assisted Living. Our system, based on a five-layer architecture, aims to identify anomalies within the home environment that may be caused by inappropriate behaviours, wrong activities, medical emergency situations, and other wasteful, nonsense or dangerous circumstances. Then, it should be able to autonomously perform high-level reasoning on the detected anomalies and consequently offer support to the user.

CHAPTER II

BACKGROUND

In this chapter, an extensive discussion of general topics related to our work is presented. The current challenges of an ageing population are discussed, as well as the domains in which solutions to these issues are being explored, the evolution of the construction industry through digital innovation and how this can help with such a sensitive issue, and the state of the art of approaches and tools that already allow similar or comparable problems to be addressed. The main topics discussed are therefore population ageing, research in the field of Ambient Assisted Living to foster an Autonomous Ageing, digital innovation in constructions and built environments, modern approaches and tools for the development of solutions supporting Ambient Assisted Living. Finally, our proposal is introduced, discussed and analysed thoroughly in the following chapters.

2.1 Ageing Issue

Population ageing is one of the four “mega-trends” that currently characterize the mankind, namely Population ageing, Population growth, Urbanization and International migration (World Population Ageing 2019, 2020). Each of these mega-trends will continue to have significant and lasting impacts on sustainable developments in the decades to come.

Focusing on the first of those, it is the result of the human successful history. Undoubtedly, the enhancement in public health and medicine, and the economic and social developments have facilitated the control of disease, prevention of injury and reduction in the risk of premature death. Besides, the increase in human longevity and the demographic transition are the engine of global population growth. The simultaneous decline in levels of fertility has triggered a continuously growing share of older people in the world population.

Population ageing phenomenon is not spreading in specific areas but characterizes the whole world. The age span of elderly people is currently increasing, and this trend is going to continue in the future. UN estimated the number of people aged 65 years or over were slightly more than 700 million in 2019, and this number is sharply going up. It is indeed projected to double by 2050, reaching 1.5 million people. Whereas the global percentage of over 65-year-old people was at just 6% in 1990, it experienced a growth by 3% within 2019 (9%) and is going to rise further to 16% by 2050 *(Fig. 1)*. This proportion outnumbered that of children aged under 5 in 2018 for the first time in history and the projections indicate that by 2050 there will be more than twice as many over 65-year-old people as children aged under 5, surpassing the number of teenagers and young adults aged from 15 to 24 years as well (World Population Prospects 2019: Highlights, 2019). Furthermore, the number of people aged 80 years or over has tripled in the last three decades, growing from 54 million to 143 million and is estimated to grow further to reach 426 million within 2050 tripling again its current value *(Fig. 2),* (World Population Ageing 2019, 2020).

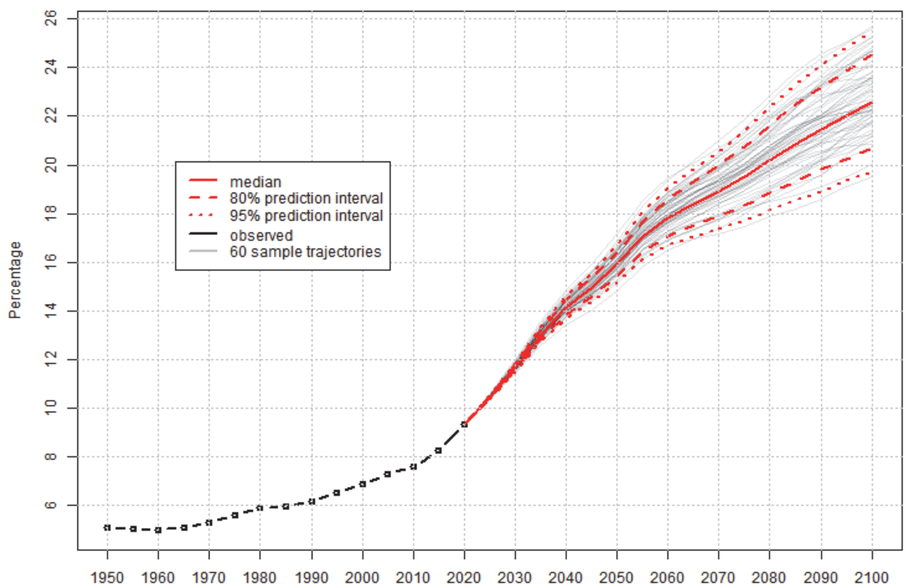


Figure 1, *Percentage of population aged 65 years or over. Estimated values and future projections* (World Population Prospects 2019: Highlights, 2019).

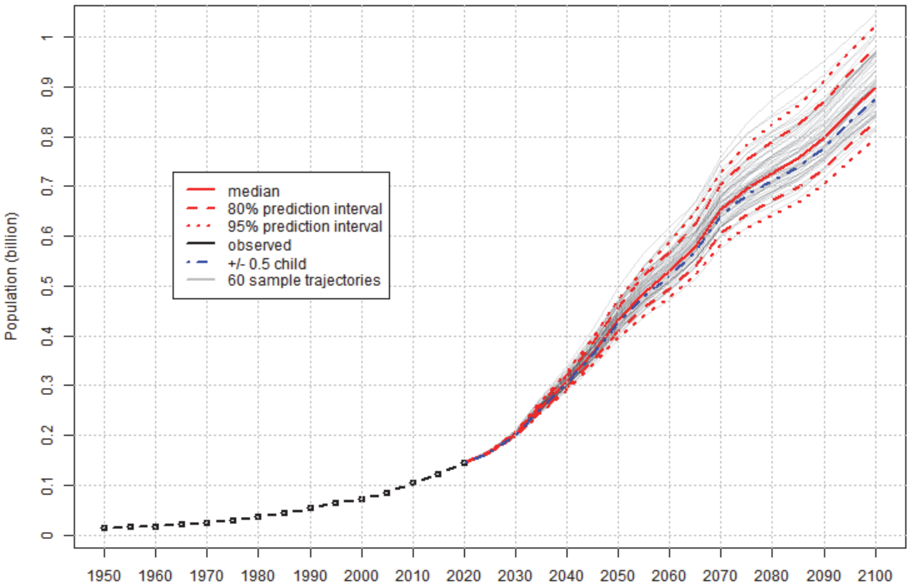


Figure 2, Number of people *aged 80 years or over. Estimated values and future projections* (World Population Prospects 2019: Highlights, 2019).

Another common indicator for monitoring changes in the age structure of the population is the Old-Age Dependency Ratio (OADR), defined as the number of persons aged 65 years or over per 100 people of working age (20 to 64 years). With declining fertility and increased longevity, the relative size of older age groups increases while that of younger age groups declines. Since 1990, the OADR has risen continuously in all regions, though its level and speed of increase have varied. Globally, in 2019 there were 16 persons aged 65 years or older per 100 persons aged 20-64. By 2050 this ratio is projected to increase to 28 per 100 *(Fig. 3)*, (World Population Ageing 2019, 2020).

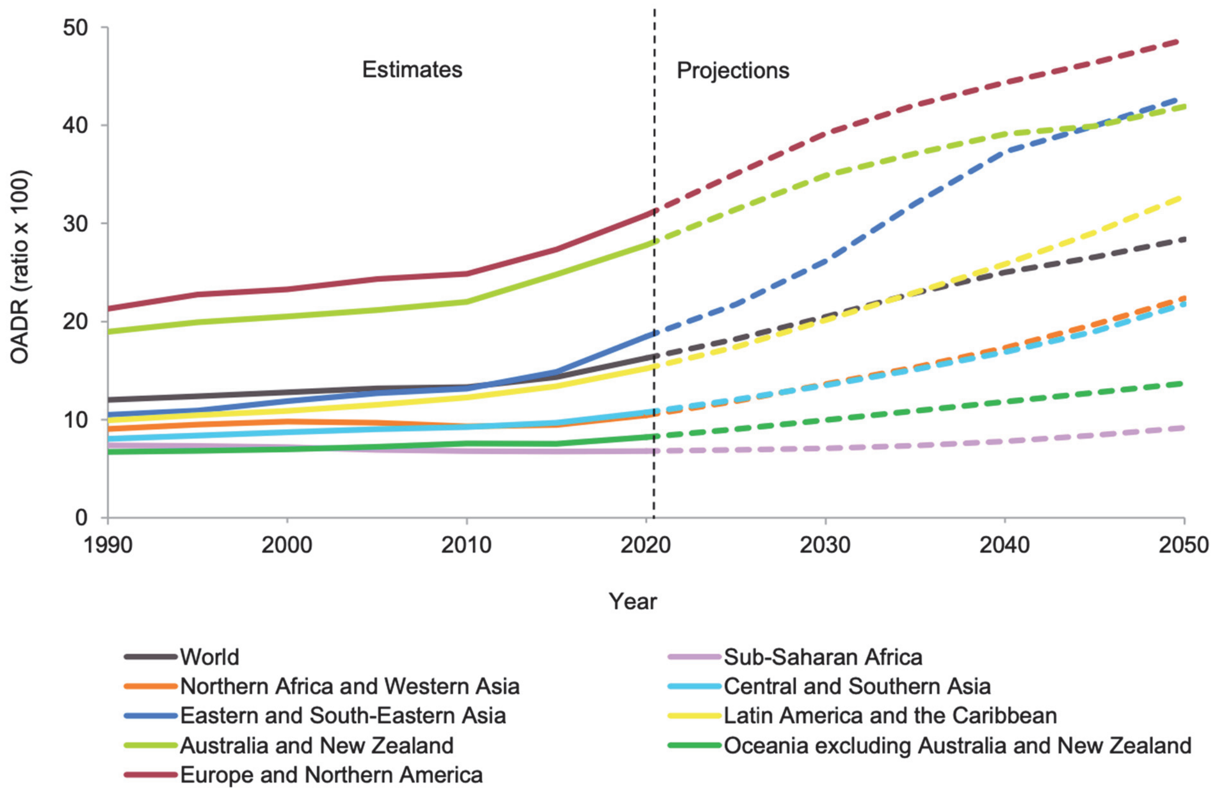


Figure 3, *Old-age dependency ratios (OADR), by region, 1990-2050* (World Population Ageing 2019, 2020).

2.1.1 Consequences of ageing population

Firstly, the impact of the ageing population has direct consequences on the elderly lifestyles *(2.1.2)*. UN has indeed estimated that there will be a higher percentage of independent older people in the future. Secondly, the increase in the number of people aged over 65 directly affects the share of the population suffering from cognitive disorders *(2.1.3)*. This is even the result of less companionship and social sharing. Furthermore, an increase in the number of pathologies such as cognitive disorders, disabilities, physical problems will indirectly become an issue for the healthcare structure *(2.1.4)*, which will likely become overloaded. In fact, structures such as hospitals, health structures, nursing homes, will have to cope with a heavier burden of patients, due to a continuously wider range of people in need of care. This will also be a burden for the countries in terms of public expenses and efficiency of public healthcare. Each of the above-mentioned issues is thoroughly analysed in the next chapters.

2.1.2 Elderly people living independently

The first significant change is determined by higher levels of elderly people living alone. As reported in (World Population Ageing 2020 Highlights: Living arrangements of older persons, 2020) the elderly tend to live independently, especially in most developed countries *(Fig. 4)*. This is a result of sufficient resources, including from pensions, personal assets or access to community founded health care. Living alone is likely to offer more privacy and control over household decisions.

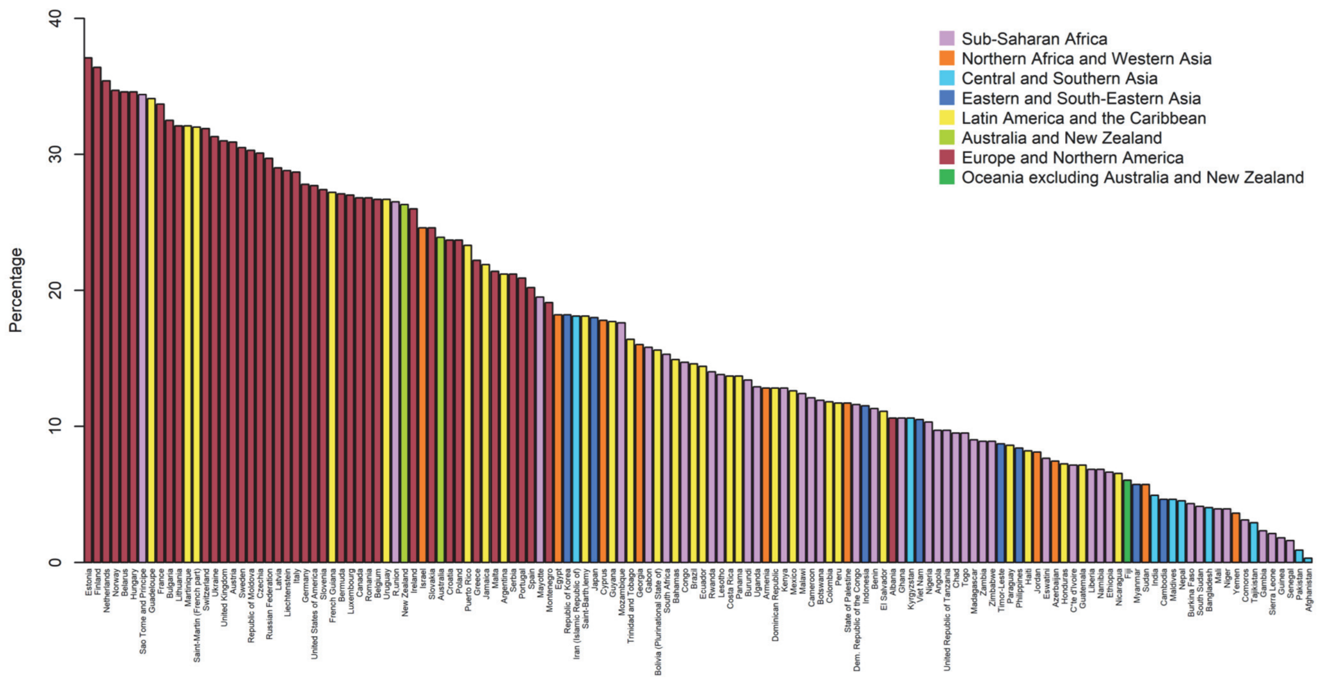


Figure 4, *Percentage of people aged 65 years or over who live alone, by country or area of residence, 2006/2015* (World Population Ageing 2020 Highlights: Living arrangements of older persons, 2020).

Nevertheless, living independently at older ages is often a consequence of either divorce or death of a partner. Less companionship and sharing are the relative side effects, therefore alone elderly people tend to rely on their children not residing with them for contact and support. They may also rely on siblings and other relatives as well as friends and neighbours whether they never had children. It is a fact that they are able to do that until either cognitive disturbances or physical problems and disabilities overcome their autonomy. At that point, the majority of them end up living with their children, while the minority is often moved to nursing homes. Besides, disabled older people were more likely to live alone in less developed countries, worsening their social isolation and deprivation (World Population Ageing 2020 Highlights: Living arrangements of older persons, 2020).

2.1.3 Cognitive disorders

At the biological level, ageing is a consequence of the impact of the accumulation of a wide variety of molecular and cellular damage over time. This leads to a steady decrease in mental and physical capacity, a growing risk of disease, and finally, death. But these changes are neither linear nor consistent, and they are only loosely associated with a person's age in years. Whereas some 70-year-olds enjoy extremely good health and functioning, other people of the same age are weak and require significant help from others. Common circumstances in older age include hearing loss, cataracts and refractive errors, back and neck pain and osteoarthritis, chronic obstructive pulmonary disease, diabetes, depression, and dementia. Moreover, elder age is characterized by the development of several complex health states that tend to occur only later in life and that do not fall into discrete disease categories. These are commonly called geriatric syndromes. They are often the consequence of multiple underlying factors and include frailty, urinary incontinence, falls, delirium, cognition disorders and pressure ulcers (World Health Organization, Ageing and health, 2018).

Specifically, when consciousness and thought processes themselves – problem solving, attention, perception, and language – are disrupted, patients are said to have cognitive disorders (Berryhill, Peterson, Jones, & Tanoue, 2012). Patients with these disorders may be disoriented, inattentive, have memory and language disturbances, and sensory irregularities including hallucinations. Cognitive disorders such as delirium, amnesia, and dementia are some of the most devastating diagnosis *(Tab. I)* for at least two reasons. Firstly, these disorders separate patients from their normal conscious states. This makes communication very challenging, and it may be difficult for caregivers to know if the patient's symptoms have been alleviated. Secondly, diagnoses such as dementia are progressive and unrelenting indicating that the patient will not return to his or her former state. Not all of those present themselves with complaints about memory or thinking as the primary symptom. For instance, frontal systems dementias may initially present as changes in behaviour or personality. In some cases, behaviours may be uncommon. Cognition may be affected, but deficits may be secondary to behavioural disturbances such as impulsivity, poor judgment, reduced initiative and planning, and impaired ability to organize. The individual is unable to engage and interact effectively with the environment, resulting in difficulties learning and conducting Activities of Daily Living (ADLs) independently (Ravdin, 2008).

As a result, someone with a frontal system dementia may be able to conduct each step of food preparation alone, but because they are unable to plan, organize, and sequence the events appropriately, they cannot prepare a meal. Likewise, they may forget to switch off appliances, close the entrance door, or undertake inconsistent actions that could lead to dangerous situations. Recognizing the presence of these types of symptoms, which may fluctuate throughout the day between periods of lucidness and confusion, can help in determining the cause of the cognitive disturbance. Diagnosing specific syndromes in the early stages of dementia is undoubtedly crucial (Ravdin, 2008). As reported in *(Tab. I)*, adequate treatments are usually based on supportive therapies, which can be supported by technological devices.

	Delirium	*Amnestic disorders*	*Dementia*
Symptoms	• Disturbed consciousness • Disorientation • Speech and language disturbances • Memory impairment for recent events	• Impaired memory • Failure to remember old or new information	• Memory impairments and other symptoms • Agnosia • Apraxia • Aphasia
Age of onset	• All ages	• Trauma: all ages • Substance abuse: older middle age	• Elderly, especially over 85 years old
Time course	• Rapid-onset • Resolution after hours or days	• Following head trauma • Prolonged onset when due to substance abuse • Permanent	• Progressive over decades • Permanent
Causes	• Medical condition • Substance induced (e.g., toxin exposure)	• Physiological general medical condition (e.g., head trauma) • Substance induced (e.g., toxin exposure)	• Disease process • Reduced blood flow • Vitamin B12 deficiency • Hypothyroidism
Treatment	• Supportive therapy • Pharmacological	• Technology (e.g., smartphones, SenseCam©) • Memory strategies • Pharmacological	• Treat symptoms • Supportive therapy

Table I, *Comparison among three types of cognitive disorder* (Berryhill, Peterson, Jones, & Tanoue, 2012).

2.1.4 Overloaded healthcare systems (after Covid-19)

Population ageing phenomenon brings with it a rising number of people requiring care, due to the fact that older people are definitely more vulnerable. The number of pathologies is going to increase dramatically while ageing, considering cognitive disorders and disabilities and further likely problems (World Health Organization, Ageing and health, 2018).

The Covid-19 pandemic has worsened the situation for two reasons principally. Firstly, a huge number of Covid-19 patients require more space and healthcare personnel than those without the virus, saturating healthcare structures, and the strict rules, which must be observed for the care, imply that healthcare workers have less time to spend with their patients. Secondly, monitoring of patients with common chronic non-communicable disease has become hard. As a matter of fact, appointment systems have struggled to assure appointments to patients, sometimes failing. Furthermore, for them it has been even more difficult to assure complete safety due to the low Doctor to Patient ratio (Jayamani, Thangaraju, Thangaraju, & Venkatesan, 2020). Despite the fact that both caretakers and patients are willing for contactless or minimal contact health services, people have not stopped from getting consultations for their acute or chronic illnesses.

In overloaded healthcare systems, both a clear and thorough communication among members of the diagnostic team and an effective interaction with patients, which is vital to patient safety, may fail. Failures or gaps in provider–patient communication may increase the likelihood of errors (MedPro Group, 2018). A diagnostic error emerges when a diagnosis is missed, inappropriately delayed or is wrong (World Health Organization, Diagnostic Errors: Technical Series on Safer Primary Care, 2016). It can be completely missed (dementia missed despite symptoms), wrong (patients told they have one diagnosis when there is evidence of another) or delayed. There may be overlaps in these classifications.

Causes of these mistakes may include cognitive errors, such as failure to synthesize the available evidence correctly. A study in one developed country found that process breakdowns most frequently involved referral problems (20%), patient-related factors (16%), follow-up and tracking of diagnostic information (15%). Further likely issues which can contribute to errors in primary care may be remoteness and travel constraints, little or no sharing of medical information, unavailability of health information resources, lack of organization of information (World Health Organization, Diagnostic Errors: Technical Series on Safer Primary Care, 2016). Thus, more than 50% of the mistakes are due to information issues.

Therefore, a change in paradigm is required more than ever. Actually, there are many healthcare providers who have already pioneered decentralized healthcare services promoting concepts like ‘Hospital on wheels’, ‘Mobile care units’ and so on. Although these are all useful as well as comfortable services, they may not be enough. Current technologies could foster decentralization even more, entering patients’ homes to exploit accurate data and gather appropriate information *(2.2.1)*. Non-overloaded healthcare structures will be able to ensure better care to patients.

2.2. Autonomous Ageing

Autonomous people are those that have the freedom to determine consistent actions to complete their own activities. *Autonomous, Healthy and Active Ageing* is the process of developing and maintaining the functional ability that enables autonomous well-being in older ages. (World Health Organization, World report on ageing and health, 2015). These capabilities are determined by the intrinsic capacity of the person (i.e., the combination of all the individual’s physical and mental – including psychosocial – capacities), the environments the individual inhabits (understood in the broadest sense and including physical, social and policy environments), and the interaction between these. Thus, the home environment has a significant influence on the development and maintenance of healthy behaviours (World Health Organization, Ageing and health, 2018). Supportive environments enable the elderly to do what is important to them in order to remain active, despite the likely losses in capacity, discussed in a previous chapter *(2.1.3)*.

The home environment provides a range of resources or barriers that will ultimately decide whether older people can engage in activities that matter to them. For instance, people suffering from cognitive disorders may not be able to follow a cooking sequence because they tend to forget its process, whereas they manage to do a single step of food preparation *(2.1.3)*. If either something or somebody in the home environment helped these people by reminding them the cooking process step-by-step, they would be capable of making a meal. In this example, the barrier may be the lack of a cooking tracking system for those who live independently. In fact, without it the elderly most probably would cook an inconsistent meal or would not start a cooking action at all, relying on other people for this specific activity. Another threat to autonomy is falls: 30% of people older than age 65, and 50% of people older than age 85, living in the community will fall at least once each year (World Health Organization, World report on ageing and health, 2015).

Environments which intrinsically help their users supporting the activities they value are called *Age-Friendly Environments*. Encouraging and easing the elderly in their ADLs, these environments help to foster an autonomous, active and healthy ageing. Thus, older people can keep autonomy, which is a core component of their well-being and has a powerful influence on their dignity, integrity, freedom, and independence. Enhancing autonomy regardless of an older person's level of capacity can be achieved through a range of mechanisms, including advanced care planning, supported decision-making and access to appropriate assistive devices *(2.2.1)*. When tailored to the individual and the environments, both of which may change over time, these systems can enable elderly people to keep the maximum level of control over their lives (World Health Organization, Global strategy and action plan on ageing and health, 2017).

2.2.1 Ambient Assisted Living

Ambient Assisted Living (AAL), as well as Information and Communication Technologies for Ageing Well (ICT), is a field of research focussed on the enhancement of the living standards of the aged people through various assistive solutions. In (Yun & Yu-Hua Gu, 2017), AAL is described as *“An ambient assisted living system creates human-centric smart environments that are sensitive, adaptive and responsive to human needs, habits, gestures and emotions. The innovative interaction between human and surrounding environment makes ambient intelligence a suitable candidate for healthcare”*. Exploiting state-of-the-art technologies, AAL develops either products or technological services as solutions to foster the environments where we live, particularly those inhabited by older people. In this regard, AAL is working on the enhancement of the autonomy of the elderly as suggested in (World Health Organization, Global strategy and action plan on ageing and health, 2017), in order to make feasible a Healthy Ageing, described in the previous chapter *(2.2).* Moreover, AAL has a cross-cutting nature which also exploits subjects such as society needs, psychology, medicine, and so on.

The aims of AAL systems are described in detail in (WHO, Regional Office for Europe, 2017). Its conceptual model for AAL application lies on three main policies *(Fig. 5)* that aim at enhancing the elderly’s activities into the home and city environments through tailored services able to foster an active and healthy ageing and, therefore, the quality of their lives:

1) *Strengthening social/informal connectivity*:
   - *Safety/security*, social care and community development starts with application areas such as those that ensure the elderly sense of safety and security in the home environment. The safety and security issue regards the emergency response with alarms (i.e., falls detectors, smoke and gas detectors, intruder alarms, etc.) directly connected with the relative assistance services.
   - *Social connection*, the second step is based on connecting the elderly to social services, events and activities through social support services. These facilities help users cope with social isolation, which usually leads to cognitive disorders.

- *Social participation*, further along this axis are proactive services that foster the elderly participation in the civic and economic life of the community. This could happen whether through formal employment (full time or part time) or volunteering.

2) *Strengthening connectivity with health system*:
   - *Vital signs*, home-based vital signs monitoring is at the base of this concept. Gathering information concerning chronic conditions at home is crucial to have appropriate diagnosis, even in primary care.
   - *Self-management*, natural extensions of vital signs devices are those which provide services enabling either a richer gathering of data at home and uploading activities to support individual empowerment to self-manage health and well-being. The tools involved are essentially interactive devices (i.e., tablets, etc.). From a health and clinical perspective additional services have evolved: monitor movement, gait, sleep quality and so on.
   - *Remote consultation*, this enables patient-centred healthcare. Screening, assessments, consultations, clinical decision-making, multidisciplinary participation are only few of the topics which would be improved by a direct online contact between patients and clinical teams.
3) *Context awareness*:
   - *Reactive*, at the base of a context-aware model are those based on predefined rules which are used to trigger alarms in order to reduce the anxiety among the elderly. Besides, they might even reduce the use of primary healthcare services.
   - *Preventive*, at this point systems are capable of gathering and analysing additional data from the user and the environment. Data are stored in order to detect behavioural changes during predefined periods of time. As a matter of fact, it is estimated that diseases such as heart attack and strokes occur 2-3 weeks after a behavioural change is registered. This enables preventive treatments.
   - *Predictive*, a predictive model is at the top of the scale concerning awareness of the contexts. These cognitive systems are increasingly sensitive and are deeply oriented towards the future ambitions of AAL, closely linked to innovation.

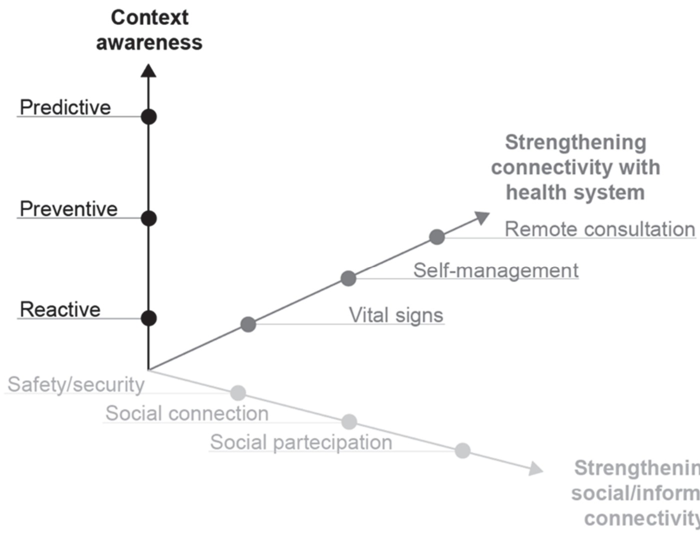


Figure 5, C*onceptual model for ICT/AAL applications* (WHO, Regional Office for Europe, 2017).

The attention from international organizations on the theme of Ambient Assisted Living demonstrates the significant impact that it might have on the lives of the older people and on the whole world. Among those, WHO and the European Commission seem the most interested in investigating this issue thoroughly. The former claims that there are five priority area for action (World Health Organization, Global strategy and action plan on ageing and health, 2017), recalled below:

1. *Commitment to Healthy Ageing.* It requires awareness of the value of Healthy Ageing and sustained commitment and action to formulate evidence-based policies that strengthen the abilities of older persons.
2. *Aligning health systems with the needs of older populations*. Health systems need to be better organized around older people's needs and preferences, designed to enhance older people intrinsic capacity, and integrated across settings and care providers. Programmes in this area are strictly associated with other works across the WHO to strengthen universal healthcare and human-centred health services.
3. *Developing systems for providing long-term care.* Systems of long-term care are needed to meet the necessities of the elderly. This requires developing, sometimes from zero, control systems, infrastructure and workforce capacity.

WHO’s work on long-standing care aligns closely with efforts to improve universal health coverage, addresses non-communicable diseases, and develops human-centred and integrated health services.

4. *Creating age-friendly environments.* This will require actions to fight ageism, enable autonomy and support Healthy Ageing in all policies and at all levels of management. These activities build on and complement WHO’s work during the past decade to create age-friendly cities and societies, including the development of the Global Network of Age Friendly Cities and Communities and an interactive information sharing platform Age-friendly World.
5. *Improving measurement, monitoring and understanding.* Dedicated research, new metrics and analytical approaches are required for a broad range of ageing issues.

Besides, the European Commission has been financing a European programme since 2008, funding innovation that supports a healthy and active ageing. The programme is called “*AAL Programme*” and is still active, arranging annual forums, workshops and challenges for researchers (AAL Association, 2008). The growth of the importance of the AAL issue has allowed the creation of State-of-the-art models. Some of them have been reviewed in the next chapter *(2.2.2)*.

2.2.2 Existing models of Ambient Assisted Living

Ambient Assisted Living domain application addresses issues related to the independence of the elderly and disabled people. If those users were able to live autonomously and comfortably for a long time, it means that they would be ageing healthy. Undoubtedly, AAL applications are not limited to home environments. However, the interests are mainly focussed on this environment. In the last years, the progress in technology has allowed the development of new AAL applications. Thus, some envision in this domain has become reality. Above all, Artificial Intelligence systems have enabled significant improvements. The scenarios of AAL services may be summarized in four main areas as suggested in (Li, Lu, & McDonald-Maier, 2015):

1. *Daily task facilitation*. Smart Home is a concept that embraces digital devices for describing the living environment and providing natural feedback and interactions with users. Smart Home systems are capable of delivering services

that facilitate the individual in its daily tasks. Those are often employed in environments where people suffering from impairment live. The services included in a Smart Home may be autonomous actuators of lighting, heating, cooling, food and drink preparation, monitoring of behaviours and risks, independent robots supporting patients in their tasks and so on. The concept of daily assistance in Smart Homes has been further analysed in the Digital Twin at a building level chapter *(2.3.1)* and Digital Twins in AAL chapter *(2.3.3)*. DTs at a building level are in fact an improvement of Smart Homes.

2. *Mobility assistance.* The quality of life of elderly people is affected by mobility issues. Movements allow them to be independent. Nevertheless, people with motor disabilities and cognitive impairments struggle with a wide range of barriers in their daily lives. As a result, they need assistance from other people. Several solutions have been studied, including assistance with planning safe paths to move between two different locations and new technological wheelchairs that exploit robotics tools. Innovative mobility devices are smart wheelchairs, Segway wheelchairs, Hybrid Assistive Limb (HAL), and so on *(Fig. 6)*.

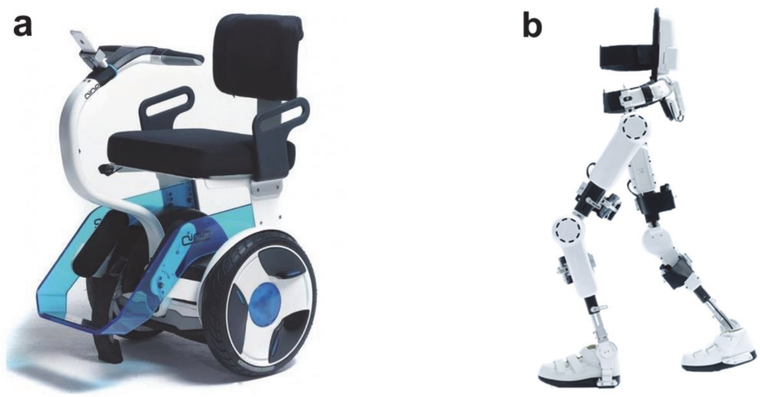


Figure 6, *(a) Segway wheelchair; (b) Hybrid Assistive Limb (HAL).*

3. *Healthcare and rehabilitation*. Health plays a key role in older ages. Thus, easing the treatment of chronic illnesses through constant health monitoring and *eCare* concept is a challenging target. Furthermore, alarms and risk assessment become significant due to the declining support from families.

Living independently has unavoidable risks and circumstances where the elderly may not be able to call for help might happen. An example of e-care system was suggested in (Sánchez, et al., 2014) within the European AAL Programme (AAL Association, 2008): “the iCarer project aimed at developing a personalized and adaptive platform to offer informal carers support by means of monitoring their activities of daily care and psychological state, as well as providing an orientation to help them improve the care provided” *(Fig. 7)*.

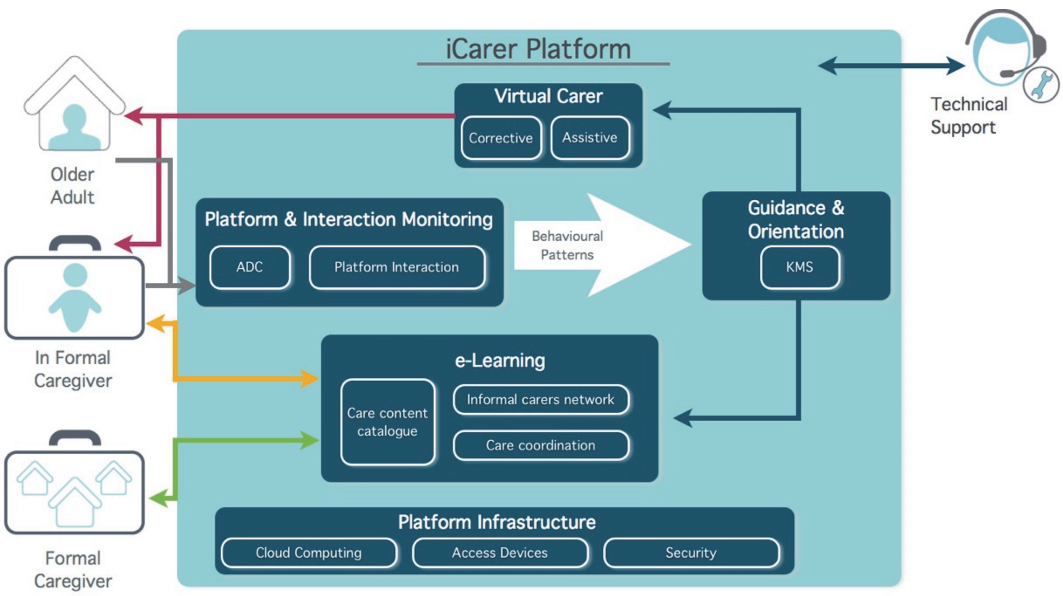


Figure 7, *iCarer Platform system* (Sánchez, et al., 2014).

4. *Communication and social inclusion*. Loneliness is one of the main causes of depression in older ages. AAL systems allow the user to stay in contact with relatives, friends and the society. Despite the fact that engaging individuals with impairments or cognitive disorders seems difficult, some AAL applications let those to remain active while staying at home. An example of that kind of application is GENIO (Gárate, Herrasti, & López, 2005). This system lets the user remain active doing supported daily activities such as reading, checking goods, creating shopping lists, following a recipe, listening to some music, watching selected photos and videos and so on. The user can communicate with the system through a microphone. Another model was designed within the NETCARITY (EU FP6) European project that aims at fostering social contacts

within the social network of older people living independently (Leonardi, Mennecozzi, Not, Pianesi, & Zancarano, 2008). In this project a prototype of e-inclusion through an interactive interface with touch-screen technology was proposed.

2.3 Digital Twins

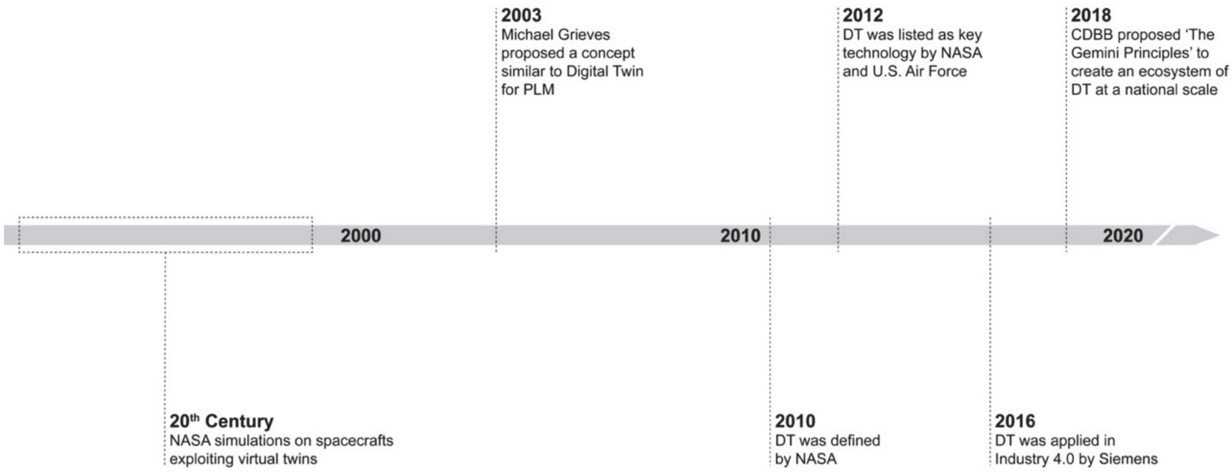


Figure 8, *Evolution of the Digital Twin concept in the last decades.*

Originally developed in manufacturing as tools to boost processes, Digital Twins (DT) are now being redefined as digital replications of assets and living entities. Actually, NASA has exploited virtual models to run simulations of its spacecrafts for decades, even in the 20th century *(Fig. 8)*. Despite that, the concept of Digital Twins became popular at the beginning of the digitalization in the manufacturing sector when Michael Grieves1 first introduced this term, in the early 2000s *(Fig. 9)*. Machinery and production systems were linked to a virtual world that gathered data exploiting sensors. The data were consequently processed through artificial intelligent methods in order to better manage the manufacturing environment. The concept of Digital Twins can be broadly applied to many technologies and is undoubtedly going to disrupt industries

1 Currently Chief Scientist for Advanced Manufacturing at the Florida Institute of Technology and Executive Director of the Digital Twin Institute. He also worked as researcher at the University of Michigan, where originally proposed the concept of "Doubleganger" for PLM in 2002, and as consultant at NASA.

beyond manufacturing. Thus, the definition of DT should be expanded to the AEC/FM sector2. By connecting the real world to its virtual twin, it is feasible to facilitate monitoring, understanding, and optimizing the functions of all physical entities (El Saddik, 2018).

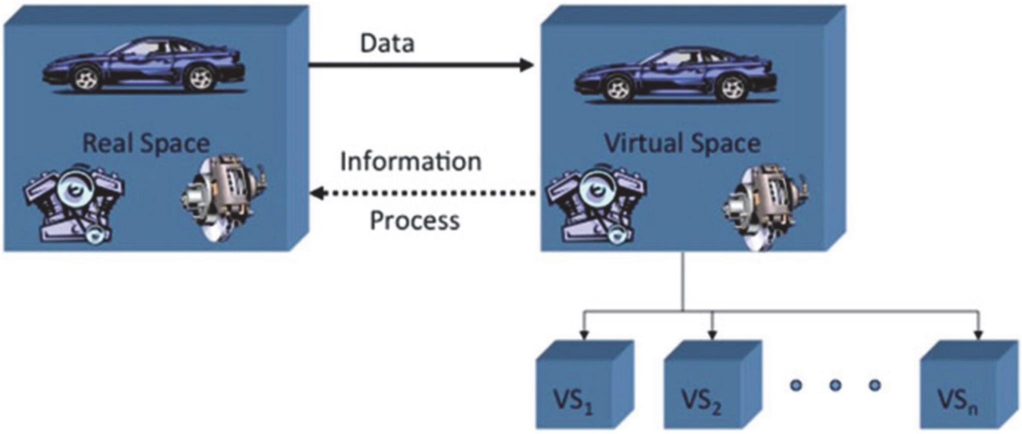


Figure 9, *Michael Grieves' original DT concept for Product Lifecycle Management in manufacturing of early 2000's. "VS" stands for Virtual Simulation* (Michael Grieves, University of Michigan, Lurie Engineering Center, Dec 3, 2002).

In the built or natural environment Digital Twins are a "realistic digital representation of assets, processes or systems" (Bolton, Enzer, Schooling, & al., 2018). The Gemini Principles3 explain that what distinguishes a digital twin from any other digital model is its connection to the physical twin. Constructed on data from the physical asset or system, a digital twin releases value mainly by supporting enhanced decision making, which creates the opportunity for positive feedback into the physical twin. The advance of the *Industry 4.0* is making cyber-physical architectures in the built environment practical, useful and affordable. Indeed, Gartner (a global research and advisory firm) predicted in 2017 that by 2021 half of huge industrial companies will use digital twins, resulting in those organizations gaining a 10% improvement in effectiveness (Pettey, 2017). In 2019, Gartner estimated that 24% of organizations that either have Internet

2 Architecture/Engineering/Construction (AEC) and Facilities Management (FM) Sector.

3 Values to guide the development and use of the information management framework and the National DT.

of Things (IoT) solutions in production or IoT projects in progress already use digital twins; another 42% plan to use twins within the next three years (Hippold, 2019). Thus, the previous predicted value seems acceptable. Hence, the impact of Digital Twins is going to be significant.

The concept of Digital Twin lies on data-driven design frameworks. One of the enabling technologies of DT is the Internet of Things (IoT). First introduced in 1999, IoT refers to the extension of the internet to the physical world through any smart device we have nowadays (smartphones, computers, sensors, actuators, and so on) and therefore is an advanced data acquisition technology. IoT can easily lead to countless benefits. Among those, appropriate data can be collected directly from the physical asset in real-time and devices can communicate and collaborate with each other. Thus, IoT and Internet allow a collection of Big Data which are crucial for creating a mirrored world of the physical one. The security of the information must be ensured to avoid data leaks and information compromises. At that point, Big Data are the input to Artificial Intelligence (AI) algorithms and analytics systems. Indeed, after extracting useful information from the gathered data, Machine Learning (ML) and Deep Learning (DL) systems are usually exploited for monitoring, predictions, simulations and feedback as well as to propose strategies in particular circumstances. The performance of the DT has to be evaluated in terms of its accuracy, resilience, robustness and costs using common evaluation methods and tests (Sharma, Kosasih, Zhang, Brintrup, & Calinescu, 2020) *(Fig. 10)*.

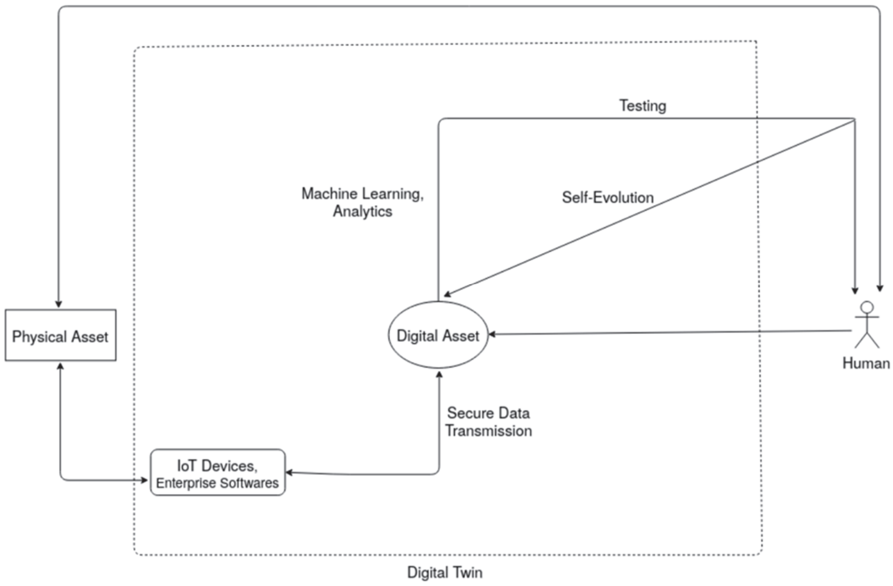


Figure 10, *Digital Twin reference framework with components and data flows* (Sharma, Kosasih, Zhang, Brintrup, & Calinescu, 2020).

In (Sharma, Kosasih, Zhang, Brintrup, & Calinescu, 2020) the principal features of DT are presented as follow:

1. *Elementary Components*:
   a. Physical Asset
   b. Digital Asset
   c. 2-way synchronized relation between physical and digital assets
2. *Imperative Components*:
   a. IoT devices
   b. Integrated Time continuous Data
   c. Artificial Intelligent system (ML and DL are the most used)
   d. Security of data and information flows
   e. Evaluation
3. *Necessary Properties*:
   a. Real-time connection between physical and digital assets
   b. Self-evolution
   c. Continuous input data to AI
   d. Continuous AI analysis
   e. Domain specific services

A number of digital twins have already begun to appear within the built environment, serving a variety of purposes depending on domains. Many more will be created. Extensive reviews of papers concerning DT technologies have been done in (Lim, Zheng, & Chen, 2020) and (Sharma, Kosasih, Zhang, Brintrup, & Calinescu, 2020). Digital twins operate at different scales *(Fig. 11)* or adopt different approaches to modelling. Among all the proposed DT frameworks in literature, The Gemini Principles are part of a wider purpose, namely, an ecosystem of DT *(Fig. 12)*. Specifically, the policies expressed aim at developing a national digital twin integrating different-scale cyber-physical systems (Bolton, Enzer, Schooling, & al., 2018). The integration of multiple DT can deliver a higher value than isolated instances. Indeed, an ecosystem of DT leads to benefit in the designing, construction and management phases of assets. Thus, the whole value chain would gain benefits, including investors, owners, managers, contractors, consultants, suppliers, tenants, or users. This provides better outcomes for the society, that will become more sustainable and resilient (buildingSMART International, 2020).

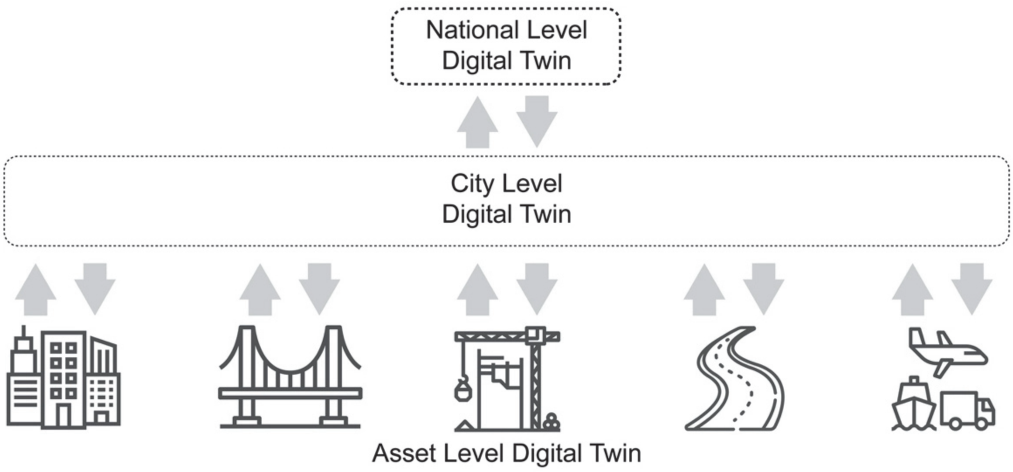


Figure 11, *Hierarchical structure of a Digital Twin ecosystem.*

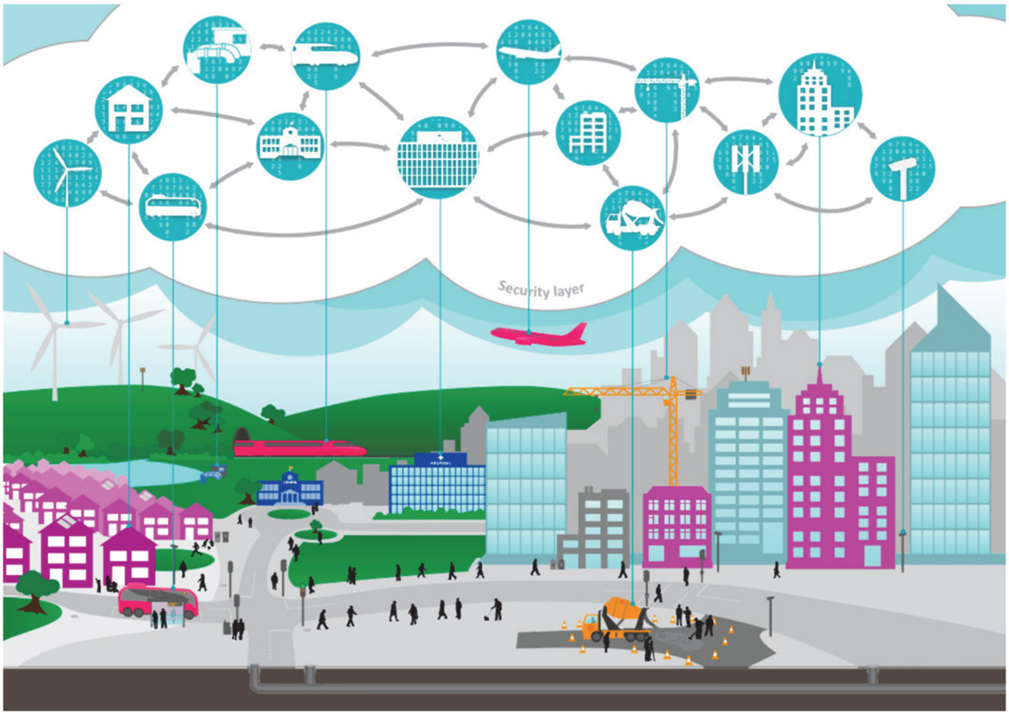


Figure 12, Ecosystem of *Digital Twins* (buildingSMART International, 2020).

Although the purpose of an ecosystem of Digital Twins seems engaging, there are some complex issues to overcome. Among those, security and integration issues seem the most challenging. The former regards the security of data and information flows between different DT that must be ensured. A likely solution may be found in blockchain technology. The latter concerns the integration of data streams at different levels of scale in the DT ecosystem and is the biggest gap to be bridged. Some interesting outcomes come from the use of the Functional Mock-Up Interface (FMI). FMI is a free standard that uses a combination of XML files, binaries and C code, supported by several tools. It enables the integration among multi-domains DT at a higher-level coordination. Nevertheless, few digital twins at present are connected or share data across organisations, sectors or geographies.

2.3.1 Digital Twins at a building level

At a building level, Digital Twins are a means of innovation in all the three phases of the AEC/FM sector (Design, construction and maintenance of buildings). In regard to the building's functionality, DT is a tool that can enhance efficiency, prevention and prediction of likely events during the whole building's life cycle. Nowadays, buildings are increasingly becoming equipped with IoT devices. A fully automatized building is best known as *Smart Building*. Digital Twin at a building level is a concept that goes beyond common smart buildings, defining cognitive and responsive environments. It describes the state of a system and its evolution over time, taking into account the users as well. The gathered data may correspond to the real state of the system, but they can also be related to a hypothetical state, which could be for example the result of simulations that use data from the Digital Twin, and produce data that are integrated in return, thus allowing to predict its future state. This description comprises static data, like the physical structure of the system, but also time-based features, like the evolution of the physical structure, and the information defining the dynamic state of the system (temperature, occupancy, and so on). When applied to Smart Buildings, the static data is the structure of the building, whether structural (building blocks, fluid sand energy networks, and so on) or functional (IoT software, space partitioning, and so on). The time-based data is of two types: the evolution of the static description over time, and the data coming from the sensors and actuators deployed in the building itself (Chevallier, Finance, & Cohen Boulakia, 2020).

In (Lu, et al., 2020), the system architecture of a building-level DT is built upon 5 layers *(Fig. 13)*:

1. *Data Acquisition Layer*, at the base of a building Digital Twin there is the tier responsible for the acquisition of information, which is a key part of the whole system. Examples of tools that enable the acquisition of data include images, videos, sensors, wireless environments, and so on. In the example below, this layer includes data from IoT devices, Asset Management, tags, sensors and other collection devices such as phones.
2. *Transmission Layer*, it is responsible for transferring the acquired data from each data acquisition tool to higher levels in order to process the information. DT architectures usually exploit either short-range coverage access network technologies such as Wi-Fi or long-range coverage technologies such as LTE,

4G, 5G. Nevertheless, new technologies such as Light Fidelity (Li-Fi) and Low-Power Wide-Area Networks (LP-WAN) seem promising as well considering the speed of transmission and their efficiency.

3. *Digital Modelling Layer,* it includes the virtual model of the building and additional information. BIM and other asset management technologies may be exploited in this tier, depending on the DT specific purpose.
4. *Data/Model Integration Layer*, it is the core of the architecture. It includes the functions required for gathering, analysing, integrating and processing the acquired data. Thus, real-time data are managed by Artificial Intelligent systems (. AI technologies update the conditions of the virtual model as the real one changes. They are also responsible for knowledge learning and, consequently, decision making. The AI engine type, embedded in a DT, depends on its purpose and domain.
5. *Service Layer*, the top level is composed of services for users. This layer is indeed responsible for the interaction between the DT and the real environment. Services in buildings may include asset management support, preventive maintenance, prediction of events and so on. In the AAL domain, services may include prevention of risks, prediction of behaviours and so on *(2.2.1)*.

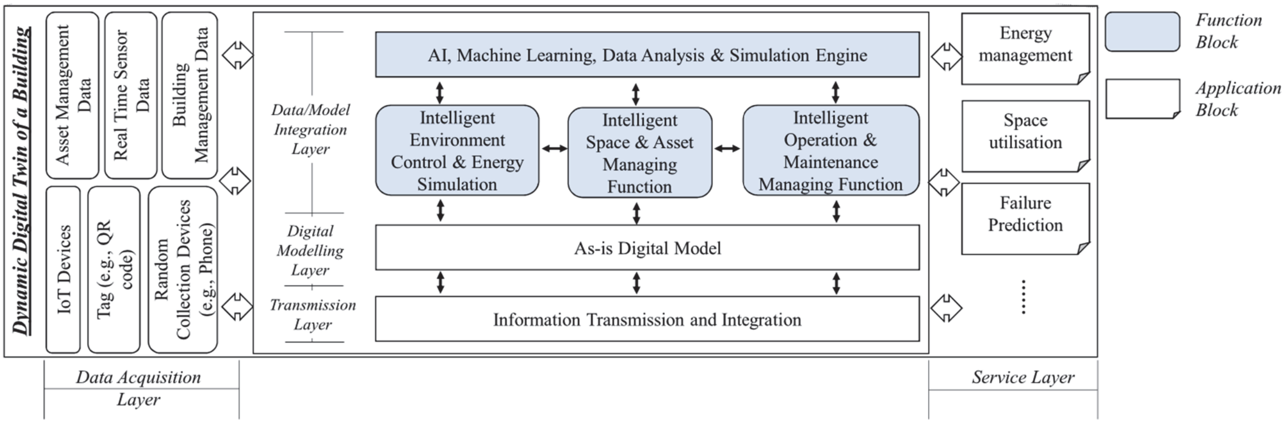


Figure 13, *System architecture of DT development at a building level* (Lu, et al., 2020).

2.3.2. Building Information Modelling

(Sacks, Eastman, Lee, & Teicholz, 2018) defines BIM as “a modelling technology and associated set of processes to produce, communicate, and analyse building models.”.

BIM is the latest and most recent stage in an evolutionary process that has deeply transformed the world of constructions over the last thirty years in all its various aspects: architectural, engineering, plant engineering, landscaping, management, and so forth. From assisted 2D and 3D design to parametric design, innovation has led to a profound change of perspective in the very way of conceiving the work in all its phases, from concept to execution and subsequent management. At the heart of this process is BIM, which is much more than a simple set of working tools, but a conceptual shift that focuses on a vision of the construction project over its lifetime, from its conception to its disposal. BIM incorporates not only information about the geometric characteristics of the building and its components, but also all the data relating to the material, economic, human and time resources, needed in the building lifetime. This philosophy represents the development of the concept of the digitization of the built environment, that is today the foundation of innovation in the construction industry. Information sharing, interoperability, and digital standards are some of the main themes on which BIM is based (AssoBIM, 2019).

2.3.2.1 From Building Information Modelling to Digital Twin

In the AEC/FM sector, Building Information Modelling (BIM) has been increasingly used as a tool to enhance interoperability and efficiency in the last decades. Targets and benefits of current BIM-enabled asset management can be summarised as follows:

- Accurate and efficient support for decision-making, monitoring and communication
- Enhanced collaboration and increased visualisation for stakeholders
- Easy retrieval and storage of maintenance, inventory, warranty, installation and operation data
- Effective management and planning of orders, activities, schedules, labour and space
- Convenient maintenance and tracking of assets
- Optimised use of utilities and materials
- Facilitation of emergency evacuation planning

Nevertheless, BIM still has limitations and issues to be addressed. Hence, in daily Asset Management (AM) is not always used properly yet, whereas it improves the efficiency of design and construction phases of the building life cycle. Those limitations of BIM in AM are not only related to organizational and technological issues, but also to a lack of a recognized standard and information. Compared with BIM, DT is more suitable for complex and comprehensive data management because DTs are built on data and have the capacities of integrating various data sources (Lu, Xie, Parlikad, Schooling, & Konstantinou, 2020). Furthermore, DT works specifically with real-time data.

As Digital Twins are platforms where different features can be integrated, BIM data could be one of them. BIM may offer data regarding buildings and their components. Those data could be consequently used by the DT to make analysis or other operations. It also enables a real-time 3D visualization of the environment. A 3D visualization offers a natural illustration that is valuable in a range of applications along the design, construction and management processes. As an intermediary of various data sources, an integration level is required, typically collating CAD, BIM, GIS, IoT data sources to the multipurpose platform (Bhargav, Buda, Nurminen, & Främling, 2018). Nevertheless, as the related data sets are typically extremely complex and heterogeneous, real-time visualization thereof has become a severe challenge. Indeed, from the literature it emerges that the BIM and IoT integration research is still in nascent stages where most proposals are at a conceptual stage. However, standardized naming formats can be used to create a link directly between the IoT and BIM data (Caroline, et al., 2020).

2.3.3 Digital Twins in Ambient Assisted Living

The growing interest in the Ambient Assisted Living domain has enabled the development of state-of-the-art DT systems for human-centred home environments. Generally, the systems monitor the users employing different types of components such as vision, object, environmental and body sensors. Sensors basically make feasible a collection of data that will be further processed in order to extract both environmental features and those regarding the user's behaviour. Advanced systems deploy multi-modal sensors (i.e., sensors that define time patterns combined with accelerometers), whereas others are based only on just one kind of signal (i.e., accelerometers). An inferring knowledge usually processes the data providing smart services to the user.

These kinds of systems are renowned as *Cognitive systems*. Those are able to learn at scale, reason with purpose and co-operate with users in a natural way. They learn and reason from the interactions with both users and environments where they are deployed, evolving with them and running *"What if?"* scenarios for predicting anomalies and behaviours. Despite the fact that the aims of DTs in AAL can vary widely (always oriented to develop WHO's three main policies *(2.2.1)), t*wo of the most crucial tasks are to detect daily living activities and abnormal activities. A review of some systems is shown below *(Tab. II)*.

Author(s)	Year	Research title	Type of framework	Features	Human activity and behaviour monitoring	Risk assessment and alarm system
Pazienza et al.	2020	Artificial Intelligence on Edge Computing: a Healthcare Scenario in Ambient Assisted Living	Healthcare architecture	Automated complex decision-making with the help of AI-based techniques directly on the Edge in healthcare scenario. The eLifeCare Platform is used to monitor and manage patients in real-time.	No	Yes
Patel et al.	2020	Real-time human behaviour monitoring using hybrid ambient assisted living framework	Generalized framework Multi-modal sensors	A generalized hybrid framework enables the development of different AAL systems with human activity and behaviour monitoring purposes using different Machine Learning and Deep Learning methods	Yes	Yes
Calderita et al.	2020	Designing a Cyber-Physical System for Ambient Assisted Living: A Use-Case Analysis for Social Robot Navigation in Caregiving Centers	Cognitive architecture Multi-modal sensors	Digital twin for caregiver centers based on CORTEX cognitive architecture enhanced with a Socially Assistive Robot.	Yes	Yes
John Fox	2017	Cognitive systems at the point of care: The CREDO program	Healthcare architecture	Supports clinical decision-making and pathway management at the point of care.	No	No
De Paola et al.	2017	An Ambient Intelligence System for Assisted Living	Multi-tier architecture Multi-modal sensors	Complete system that enable automatic environmetal changes depending on user satisfaction (i.e., temperature). User condition allows the system to infer anomalies exploiting Dynamic Bayesian Networks	Yes	Yes
Yared et al.	2015	Cooking risk analysis to enhance safety of elderly people in smart kitchen	Alarm framework	Enables kitchen safety for elderly people proposing an insightful cooking risk analysis and assessment to prevent potential risks.	No	Yes
Suryadevara et al.	2013	Forecasting the behavior of an elderly using wireless sensors data in a smart home	Multi-tier architecture Wireless Sensor Network	Low-cost, robust, flexible and data driven intelligent system. Data from appliances define behaviours which may be either regular or not.	Yes	Yes

Table II, *Review of some DT frameworks within AAL environments.*

2.4 Artificial Intelligent systems in Ambient Assisted Living

Since the value of AAL has been recognized, Artificial Intelligence methods (AI) have been used in that domain. The impact of AI in AAL is significant due to the fact that those systems are able to learn from past experiences and then reason as a human in a logical way. The AI algorithms deployed in AAL environments are usually used for detecting objects and behaviours, or inferring activities (sleeping, eating, drinking, cooking, and so on) and specific actions (cutting, taking, and so on), or detecting anomalies. Thus, the purpose of these systems may vary widely, but are all oriented to analyse and process data to enhance and support the user's daily life as well as its safety. Machine Learning (ML) and Deep Learning (DL) algorithms are the most popular AI systems in the AAL domain.

As previously explained, the inputs of AI systems are generally data coming from the *Data Acquisition Layer* through the *Transmission Layer (2.3.1)*. Nevertheless, depending on the position and the number of AI models deployed in a system, their inputs are of two types:

1. Data from sensors. They may be either raw data or pre-processed data (pre-processing is often used to remove noise and to do adjustments, fusion or transformations).
2. Data from preceding AI systems.

In the first case, AI usually performs low-level reasoning (e.g., recognize an object or an activity). On the contrary, the second case al generate high-level decisions from low-level information yielded by preceding systems.

2.4.1 Activity Recognition

Activity Recognition (AR) of humans is crucial in offering a wide range of services to users in Ambient Assisted Living environments. One of the main reasons for its importance is that the wellness of older people is ensured if they are able to perform daily activities independently at regular intervals of time. The unpredictable nature of inhabitants, especially those suffering from cognitive disorders, makes this task definitely challenging. There are two main approaches, namely data-driven and knowledge-driven (Rafferty J. , Nugent, Liu, & Chen, 2017).

Data-driven approaches consist in exploiting datasets to learn models of daily activities through statistical and probabilistic methods. Datasets are collections of gathered data that usually come from sensor's activation signals or images captured from cameras. Two data mining and Machine Learning methods are usually used as learning systems: generative and discriminative. The main benefits of these approaches are that they enable the modelling of uncertainty and time-based parameters. On the contrary, they require a suitable large dataset to learn from. Furthermore, the scalability of these models is limited due to the close relationship between a particular dataset and the environment where it has been created. Examples of data-driven approaches are methods such as Hidden Markov Models, naïve Bayes and Bayesian Network *(2.4.2.1)* (generative approaches), and Artificial Neural Networks *(2.4.1.1)* (discriminative approaches).

Knowledge-driven approaches use domain knowledge, which is an intuitive record learned through people's experiences, and a priori heuristics as the foundations to produce activity models. In general, this kind of method is either logical or ontological. The first one expresses ADLs through logical structures using knowledge representation formalisms. The combination between logical methods and knowledge-based inference is the key to support the activity recognition objective. A positive aspect of logic-based approaches is that they do not require a training phase with datasets. Consequently, logic-based approaches are scalable and can be easily used in environments which differ from those where a specific dataset has been produced. Nevertheless, these methodologies find difficulties in representing uncertainty. Ontology-based approaches, such as those in (Rafferty J. , Nugent, Chen, & Liu, 2015) and (Rafferty J. , Nugent, Liu, & Chen, 2017), organise knowledge using hierarchy of concepts and classes, that may have sub-properties, relationships and constraints. Ontologies allow a simple reuse of activity representations, overcoming the difficulties encountered in logical approaches. Uncertainty is an issue in these methods as well.

A review of the latest activity recognition models is shown below *(Tab. III). Knowledge-driven approaches are less in number due to the need for domain knowledge. Input data may vary depending on the method chosen. Currently,* several types of sensors have been exploited (gyroscopes, accelerometers, sound sensors, etc.). The latest and most accurate systems are based on visual data. Multi-modal sensors may also be used for this task, combining different input data in order to have a higher accuracy. Nevertheless, visual-information based daily activity recognition is quickly evolving and it will likely achieve higher performances.

Methods reviewed	**Advantages**	**Disadvantages**	**Input**	**D-driven**	**K-driven**	**Examples**
Neural Networks	• Cope with data uncertainty • Multi-modal input available • Allow a self-optimization • Open datasets • No domain knowledge required • Sensors are not mandatory (just a camera) • Codes available • Multi-person settings available	• Must be trained • Limited scalability • Significant amount of data • Privacy issues	*(see Chapter. 2.4.1.1)*	X		*(see Chapter. 2.4.1.1)*
Bayesian Networks	• Cope with data uncertainty • Multi-modal input available • Allow a self-optimization • Open datasets • No domain knowledge required • Enable context data fusion • No privacy issues	• High level of complexity • Limited scalability • Huge number of sensors required	Readings from sensors	X		De Paola et al. (2017) Xiao & Song (2017)
Conditional Random Fields	• Cope with data uncertainty • Multi-modal input available • No domain knowledge required • Enable context data fusion • No privacy issues	• Must be trained • Limited scalability • Huge number of sensors required	Sensor's time sequences	X		Bharti et al. (2019)
Intelligent Agents	• Scalability, reusability • No training required • Multi-modal input available • No privacy issues	• Domain knowledge required • Difficulties in representing data uncertainty • Huge number of sensors	Readings from sensors		X	Rafferty et al. (2015) Rafferty et al. (2017)
Temporal pattern-based approaches	• Several ML classification methods may be used (Support Vector Machines, Naive Bayes, k-Nearest Neighbors, etc.) • Multi-modal input available • No domain knowledge required • Open datasets • No privacy issues	• Huge number of sensors required • Must be trained • Limited scalability	Sensor's time sequences	X		Liu et al. (2015)
K-means clustering	• Multi-modal input available • Little training time • Small computational complexity • No domain knowledge required • No privacy issues • Automatic creation of custom-made datasets • Scalability	• Suitable for anomaly detection (fall, etc.) • Must be trained • Huge number of sensors required	Readings from sensors	X		Zhao et al. (2018)

Table III, *Review of various activity recognition methods (D-driven stands for Data-driven approach, K-driven stands for Knowledge-driven approach).*

2.4.1.1 Artificial Neural Networks

Artificial Neural Networks (ANN), usually called Neural Networks (NN), are a subset of Machine Learning and are at the heart of Deep Learning algorithms. They are computing systems defined by neurons and edges with their relative weights. Inspired by the biological neural networks that constitute brains, these systems have to be pre-trained to work. Training is processed using datasets. The networks are composed of hidden layers that compute output through some non-linear functions and various transformations *(Fig. 14)*.

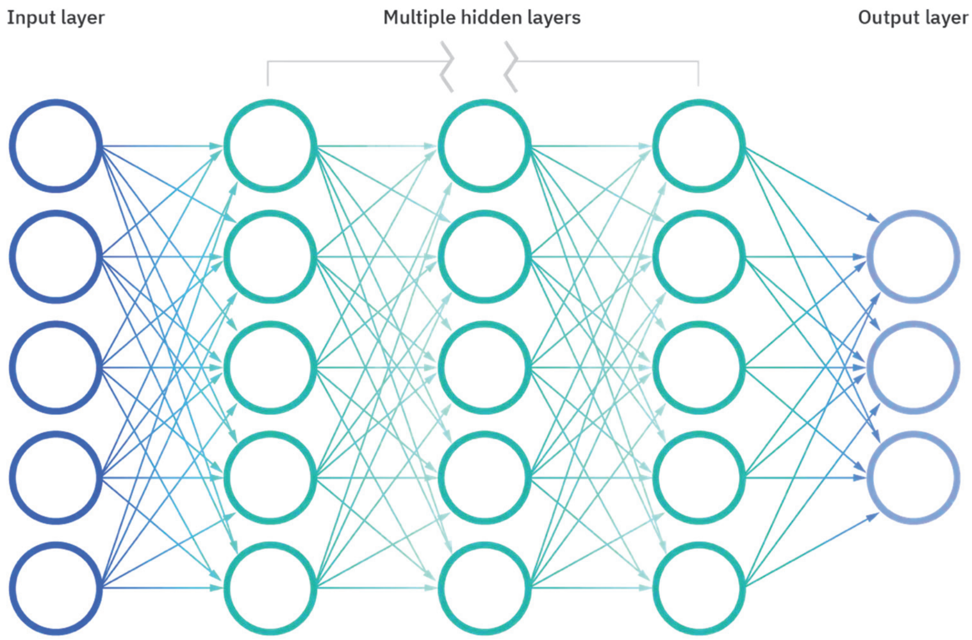


Figure 14, *System architecture of a Neural Network* (IBM Cloud Education, 2020).

Neural Networks such as R-CNN and Fast R-CNN were first used for object recognition task. Girshick et al. (Girshick, Donahue, Darrel, & Malik, 2014) first proposed a Convolutional Neural Network combined with Region proposals called R-CNN in 2014 *(Fig. 15)*. Their system outperforms already existing algorithms by more than 30%. However, its training time required more than 84 hours and its test time was extremely far from real-time (47 seconds per image).

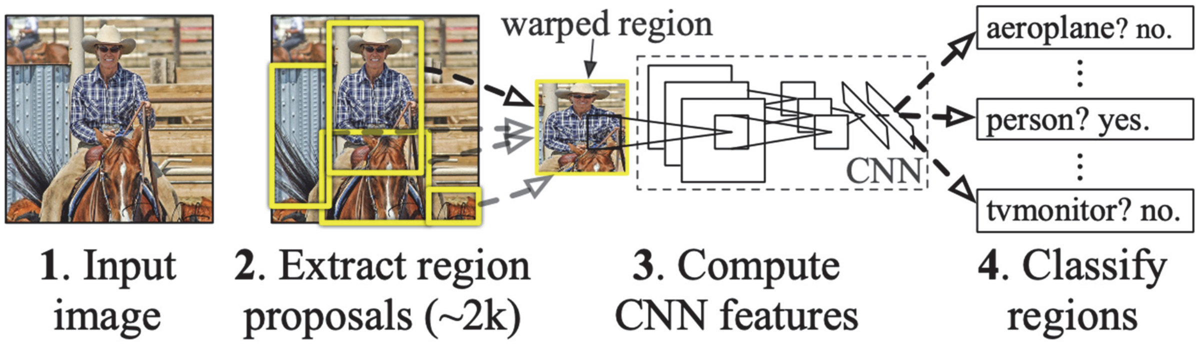


Figure 15, *Region proposals with CNN features* (Girshick, Donahue, Darrel, & Malik, 2014).

Fast R-CNN allowed the achievement of better outcomes. Using the same selective search method of its predecessor, this system firstly feeds the CNN directly with images, generating a convolutional feature map, and then identifies regions. Avoiding initial region proposals, Fast R-CNN could be trained in 8.75 hours and test time decreased at 2.3 seconds. An implementation of Fast R-CNN, namely Faster R-CNN, succeeded in real-time testing (0.2 seconds per image) by removing region proposals. You Look Only Once (YOLO) end-to-end system achieved interesting outcomes in 2016 (Redmon, Divvala, Girshick, & Farhadi, 2016) *(Fig. 16)*. YOLO first version *"frames object detection as a regression problem to spatially separated bounding boxes and associated class probabilities"*, outperforming previous methods processing 45 frames per second. Throughout the last years, this operation has therefore become incredibly fast thanks to disruptive systems. Indeed, object detection could be now processed in real-time even on our smartphones.

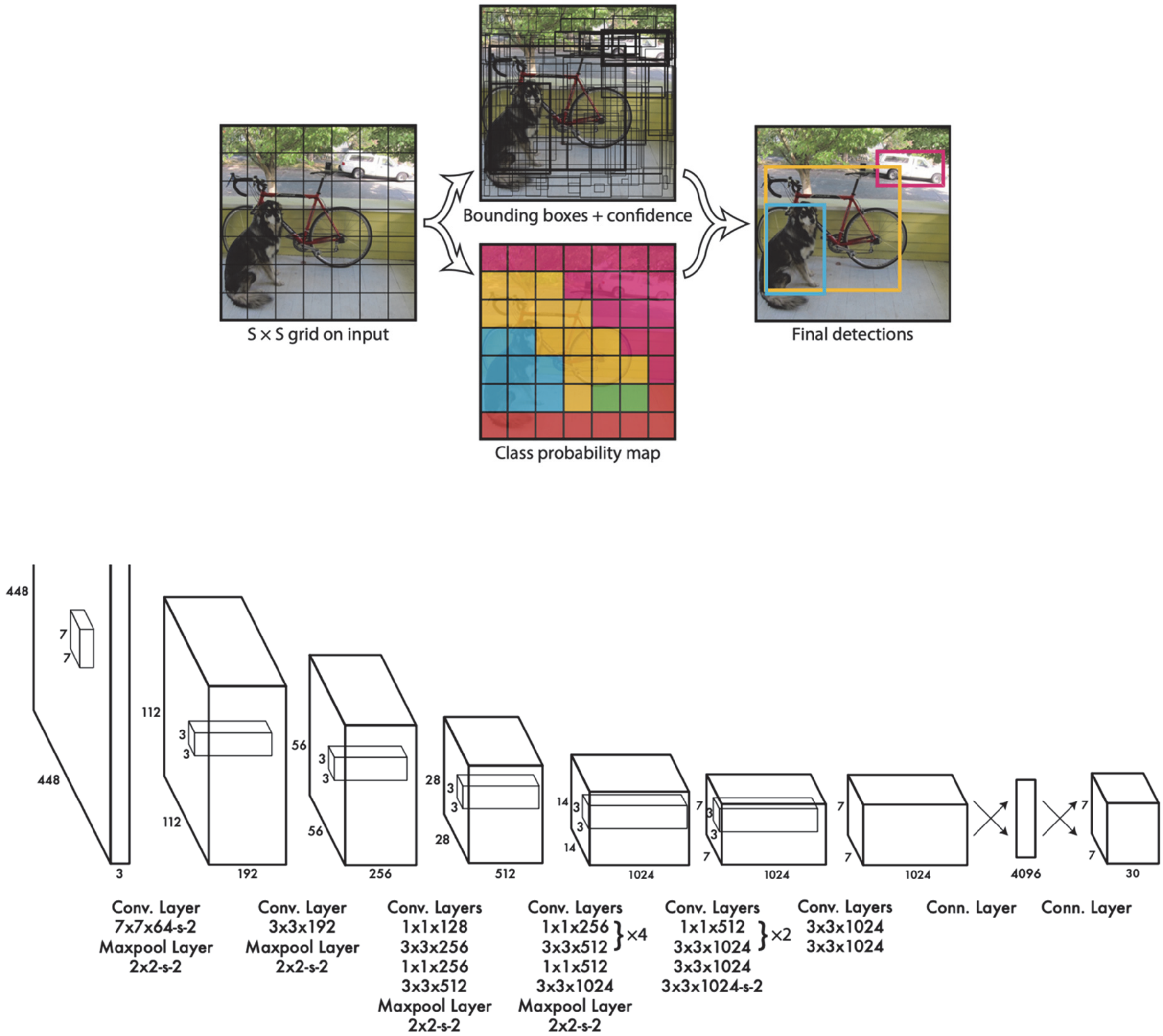


Figure 16, *YOLO object detection model and architecture* (Redmon, Divvala, Girshick, & Farhadi, 2016).

Although Neural Networks are still exploited in object recognition tasks as they are breakthrough systems in that domain, they have been leveraged for improving activity recognition systems since 2017. Despite the fact that the application of visual data in activity recognition is extremely recent, nowadays there are a huge number of ground-breaking studies that try to reduce bias and complexity of latest systems. Though, most recent system architectures are always based on:

- *Recurrent Neural Network (RNN)*
- *Convolutional Neural Network (CNN)*
- *Graph Convolutional Network (GCN)*

Before those methods emerged in 2017, Convolutional and Recurrent NN were used to recognize activities completed by users mining temporal patterns. A work that goes along with this theory was proposed in (Ordóñez & Roggen, 2016), which outperformed traditional heuristic processes and some CNN baselines as well. However, they still did not make use of visual data. They proposed a network called *DeepConvLSTM* based on convolutional and Long short-term memory recurrent units (LSTM) that *"explicitly models the temporal dynamics of features activation"*. The input to the network consists of data sequences defined by short time series. The time pattern is directly extracted from the data coming from body sensors. Four Convolutional layers transform input data in a feature map. The feature map is then processed in Recurrent dense layers built on LSTM units. The LSTM unit, which extends RNN with memory cells, enables the system to learn temporal relationships on long time scales and its internal memory can be updated, erased or read out every time. Finally, the outputs are obtained through a Softmax layer which yields a class probability distribution for every single time step *(Fig. 17)*.

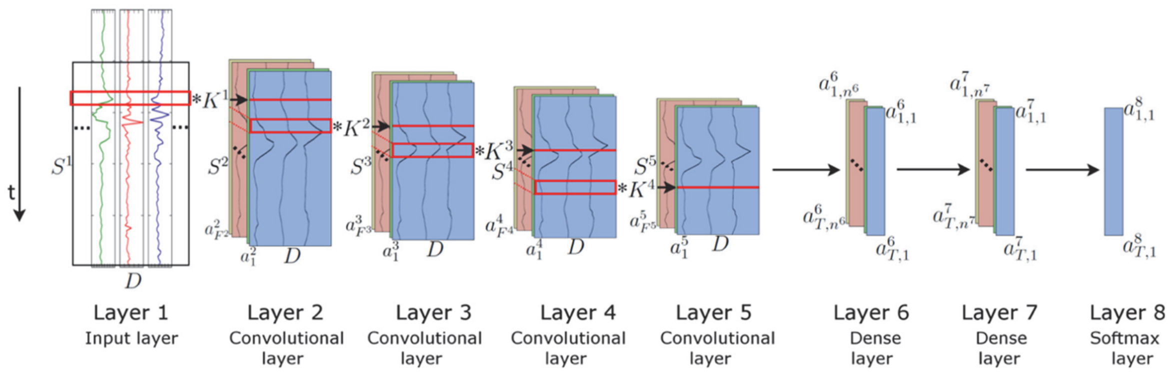


Figure 17, *Architecture of the DeepConvLSTM framework for activity recognition* (Ordóñez & Roggen, 2016).

The emergence of highly-accurate systems and affordable devices for recording 3D human skeleton data enabled researchers to make a good step forward in terms of efficiency in activity recognition. Skeleton tracking, also referred to as Articulated human pose tracking, records the trajectories of the user's skeleton joints, capturing information required to understand the activities undertaken. 3D skeleton data are valuable and comprehensive for summarizing various human dynamics. Compared with other methods (e.g., RGB images, depth maps), skeleton-based activity recognition has the advantage of being more robust to noise such as background and irrelevant objects. Furthermore, low-cost capture systems have spread in the last years and, thus, large-scale 3D skeleton datasets have been made available. Nevertheless, systems for 3D skeleton tracking include:

- Nuitrack 4
- OpenPose 5
- Cubemos Skeleton Tracking 6

Devices that enable skeleton tracking include:

- Intel RealSense 7
- OpenCV Oak series 8

Most popular datasets include:

- NTU RGB-D 9
- Kinetics 10

Two parallel practises are usually followed to detect and classifying activities through 3D skeleton:

a) The first one consists in transforming skeleton video sequences in RGB images. Examples can be found in (Li, Chen, Yucheng, Yuchao, & Mingyi, 2017), (Li, Zhong, Xie, & Pu, Skeleton-based Action Recognition with Convolutional Neural Networks, 2017), (Ke, Bennamoun, An, Sohel, & Boussaid, 2017), (Li, Zhong, Xie, & Shiliang, Co-occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation, 2018).
   CNNs are trained with datasets containing RGB images through the backpropagation algorithm which aims to reduce a cost function that quantifies the error of the model on the dataset used. Backpropagation is used to compute the gradients of the error with respect to all the weights in the network and gradient descent algorithm updates all parameters to minimize the output error

4 www.nuitrack.com
5 https://github.com/CMU-Perceptual-Computing-Lab/openpose
6 www.cubemos.com/skeleton-tracking-sdk
7 www.intelrealsense.com
8 https://store.opencv.ai/
9 http://rose1.ntu.edu.sg/datasets/actionrecognition.asp
10 https://deepmind.com/research/open-source/kinetics

(Laraba, Tilmanne, & Dutoit, 2019). When the amount of training data is not sufficient to adjust the parameters of the system, transfer learning and fine-tuning are used. In these situations, the models are previously pre-trained on large-scale datasets and then on the desired one. RGB information usually symbolizes movement along axis (x, y, z). An example of a simple transformation unit has been proposed in (Laraba, Tilmanne, & Dutoit, 2019):

$$\begin{cases} r_i(f) = 255 * \dfrac{x_i(f) - min(X)}{max(X) - min(X)} \\ g_i(f) = 255 * \dfrac{y_i(f) - min(Y)}{max(Y) - min(Y)} \\ b_i(f) = 255 * \dfrac{z_i(f) - min(Z)}{max(Z) - min(Z)} \end{cases} \quad (1)$$

In this system, $(x_i(f);\ y_i(f);\ z_i(f))$ are the 3D coordinates of the joint i at frame f. $min(X), max(X), min(Y), max(Y), min(Z), max\ (Z)$ are minimum and maximum values for all joints on the three axes. Consequently, $(r_i(f);\ g_i(f);\ b_i(f))$ are the normalized values, corresponding to the three channels of the RGB space. At that point, by stacking all converted joints on subsequent frames we obtain an image-like representation of spatial and temporal features of the skeleton *(Fig. 18)*. Once the transformation is performed, pre-trained Convolutional Neural Networks are fed by those RGB images in order to extract feature maps to classify the activity undertaken by the user *(Fig. 19)*.

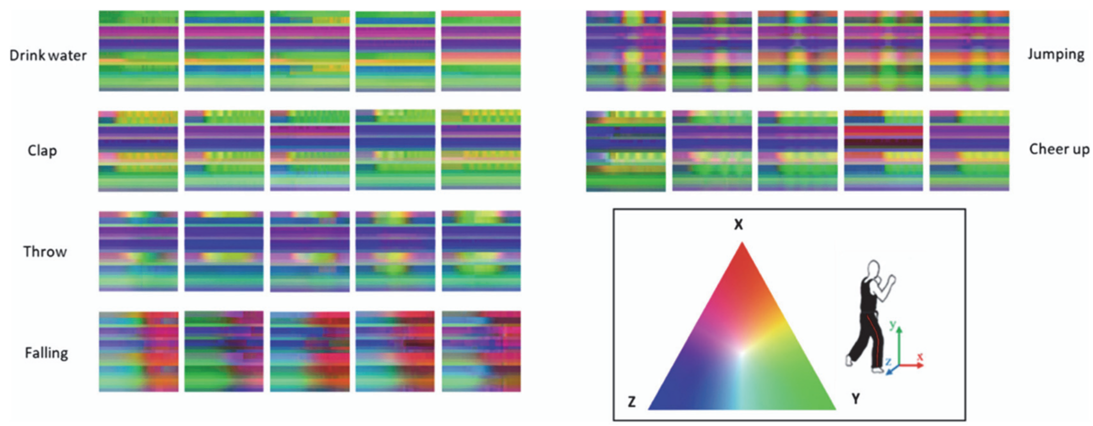


Figure 18, *Example of motion capture sequences represented in RGB images just after being converted from 3D coordinates. Vertical axes represent the skeleton joints and horizontal axes*

represent frames. The image has been resized to a square in order to be easily fed in CNN (Laraba, Tilmanne, & Dutoit, 2019).

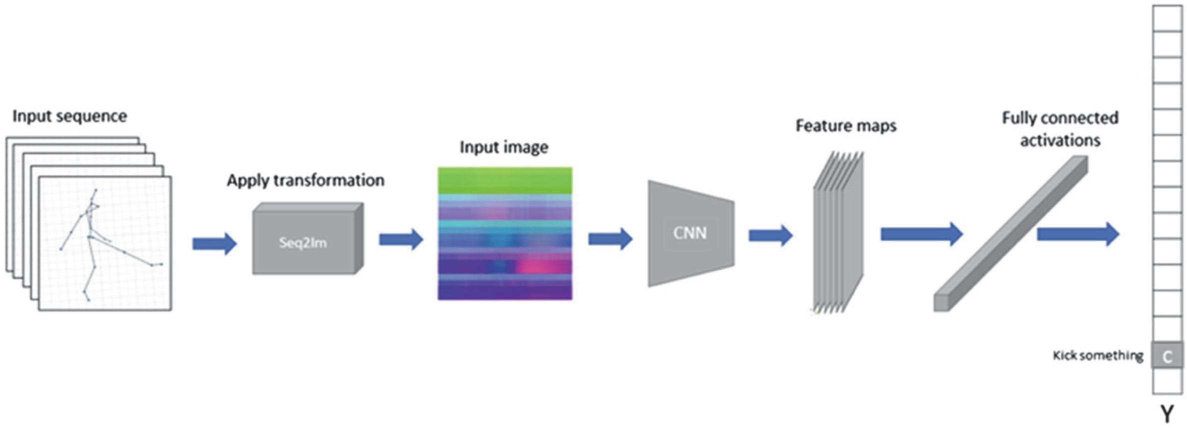


Figure 19, *End-to-end baseline proposed in Laraba et al. for skeleton-based action recognition normalizing 3D joints coordinates to RGB data* (Laraba, Tilmanne, & Dutoit, 2019).

b) On the contrary, the second practice directly uses 3D skeleton coordinates relative to skeleton joints. Thus, in this case the baseline is built upon Graph Convolutional Networks which are a generalization of common CNNs. Examples can be found in (Yan, Xiong, & Lin, 2018), (Liu, Zhang, Chen, Wang, & Ouyang, 2020). GCNs usually exploit skeleton joints' spatial-temporal information and detect differences among subsequent time frames. Skeleton joints become thus graph nodes and connectivities among joints and time become graph edges. The joint coordinate vectors are then the input of the GCN. Yan et al. proposed a Spatial-Temporal GCN system composed of 9 layers of graph convolution operators that act similarly to convolutional layers. Finally, the output results from a Softmax classifier, which yields the probability that an activity has been pursued by the user *(Fig. 20)*. Their network has been trained in an end-to-end manner using a backpropagation algorithm on 3D skeleton datasets (e.g., Kinetics). Whereas the former approach may find issues due to weak generalization, the latter allows a stronger one.

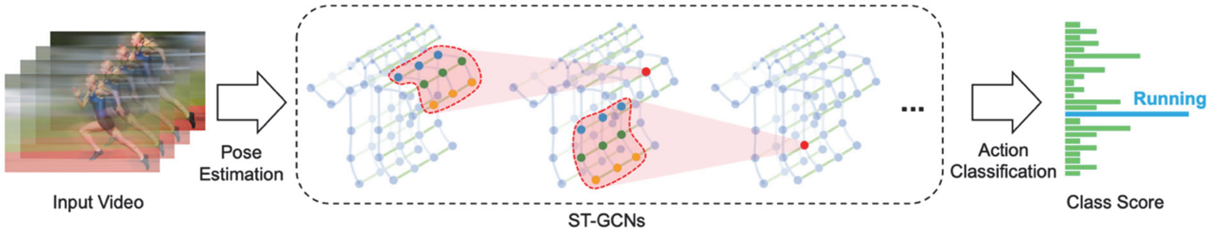


Figure 20, *End-to-end baseline proposed in Yan et al. for skeleton-based action recognition based on graphs* (Yan, Xiong, & Lin, 2018).

These two methods have been examined in depth in many papers to define state-of-the-art activity recognition systems based on Neural Networks. Some systems, such as the one proposed in (Mao, et al., 2021), rely on both approaches in order to have higher accuracy.
A review of papers regarding those systems is shown on the next page *(Tab. IV)*.

Author(s)	Year	Network	Acronym	Main Features	RGB-D	3D Skeleton	Dataset	Accuracy (%)
Song et al.	2020	Richly Activated Graph Convolutional Network v2.0	RA-GCN	Multi-stream GCN responsible for learning discriminative features from the joints less activated by preceding streams. Improve robustness overcoming incomplete or noisy skeleton.	X		NTU RGB+D 60 X-Subj NTU RGB+D 60 X-View NTU RGB+D 120 X-Subj NTU RGB+D 120 X-Set	87.3 93.6 81.1 82.7
Song et al.	2020	Residual Graph Convolutional Network v1.0	Res-GCN	Baseline based on GCN: Early fused Multiple Input Branches (MIB) + Residual GCN, inspired by ResNet architecture in CNN, with bottleneck structure + Part-wise Attention (PartAtt) block.	X		NTU RGB+D 60 X-Subj **NTU RGB+D 60 X-View** **NTU RGB+D 120 X-Subj** NTU RGB+D 120 X-Set	90.9 **96.0** **87.3** 88.3
Das et al.	2020	Video-Pose Network	VPN	The 2 key components of this VPN are a spatial embedding and an attention network. The first one aligns 3D and RGB data. The second one can discriminate similar actions leveraging weights.	X	X	**NTU RGB+D 60 X-Subj** **NTU RGB+D 60 X-View**	**93.5** **96.2**
Liu et al.	2020	Multi-Scale spatial-temporal graph convolutional operator	MS-G3D	Multi-Scale aggregation scheme disentangles the importance of nodes in different neighbourhoods for effective long-range modelling. The proposed G3D module leverages dense cross-spacetime edges as skip connections for direct information propagation across the spatial-temporal graph.		X	NTU RGB+D 60 X-Subj **NTU RGB+D 60 X-View** NTU RGB+D 120 X-Subj **NTU RGB+D 120 X-Set** **Kinetics Skeleton 400 Top-1**	91.5 **96.2** 86.9 **88.4** **38.0**
Yan et al.	2018	Spatial-Temporal Graph Convolutional Network	ST-GCN	Learns automatically spatial-temporal pattern from data. The Graph Convolutional Network enable a stronger generalization capability.		X	"NTU RGB+D 60 X-Subj NTU RGB+D 60 X-View" Kinetics Skeleton 400 Top-1	81.6 88.8 31.6
Chao Li et al.	2018	Hierarchical Co-occurrence Network	HCN	Co-occurrence features are learned with a hierarchical methodology, in which different levels of contextual information are aggregated gradually. Two-stream network.	X		NTU RGB+D 60 X-Subj NTU RGB+D 60 X-View	86.5 91.1
Chao Li et al.	2017	Convolutional Neural Network	CNN	A transformer module select only important joints automatically. These joints are fed directly in a 7-layer convolutional network for action detection and classification.	X		NTU RGB+D 60 X-Subj NTU RGB+D 60 X-View PKU-MMD X-View PKU-MMD X-Subj	83.2 89.3 90.4 93.7
Ke et al.	2017	Convolutional Neural Network	CNN	A skeleton sequence is transformed into 3 clips consisting of grey images. The generated clips are then fed to a deep CNN model to extract features used in a Multi-Task learning network (MTLN) for action recognition.	X		NTU RGB+D 60 X-Subj NTU RGB+D 60 X-View	79.6 84.8
Bo Li et al.	2017	Convolutional Neural Network	CNN	Skeleton-based video image mapping, which encodes a skeleton video to a color image in a temporal preserving way, and an end-to-end trainable fast skeleton action detector based on image detection.	X		PKU-MMD X-View PKU-MMD X-Subj	94.2 54.8

Table IV, *Review of papers concerning state-of-the-art Neural Network system architectures for Activity Recognition task.*

2.4.2 Situation Awareness

M.R. Endsley11 defined SA in dynamic environments as *"the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future"* (Endsley, 1995). Basically, it is about comprehending what is happening in the environment to predict future situations. The concept of SA is usually broken down in three sub-concepts: perception of the elements in the environment, comprehension of the situation, and projection of future status *(Fig. 21)*. Hence, SA enables high-level reasoning.

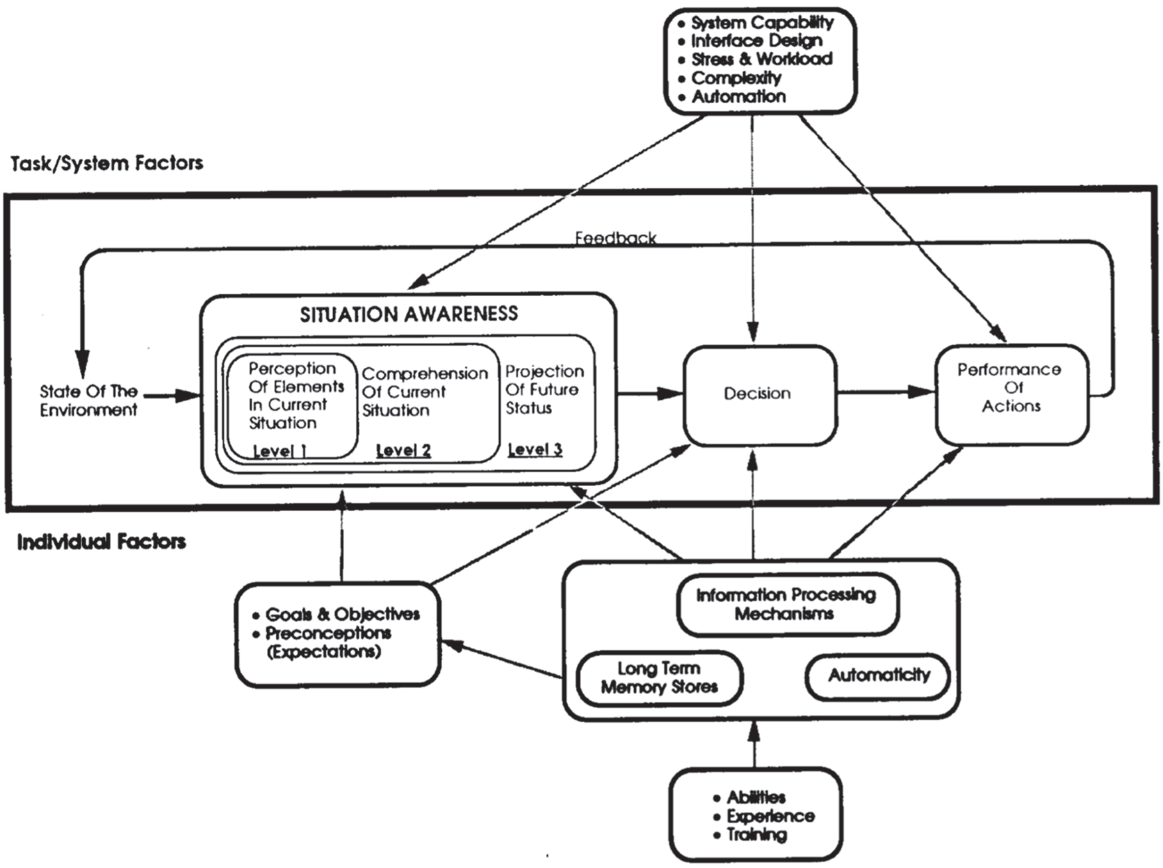


Figure 21, *Endsley's model of situation awareness in dynamic decision making* (Endsley, 1995).

11 First Human Factors engineer, served as Chief Scientist of the United States Air Force. Founder of the *Journal of Cognitive Engineering and Decision Making*.

SA becomes relevant in environments characterized by high levels of uncertainty and complexity. Its capability to reduce or avoid at all human errors makes it valuable in domains such as healthcare, aviation, and so on (Nini, 2020). Thus, SA is perfectly suitable for AAL scenarios. Digital Twins are by definition containers of real-time knowledge about a real asset and SA could be a relevant feature of a DT. In AAL environments where DT systems are deployed, an SA agent might be expressed through a high-level *Reasoner.* A Reasoner enables high-level decision-making exploiting appropriate information regarding the environment, the user and the activity performed. A simple Reasoner is deployed in the system built in (De Paola, et al., An ambient intelligence system for assisted living, 2017). Their module is a rule-based reasoner, influenced by the output of an activity recognition module, the conditions of the user and his or her satisfaction, and takes decisions such as turning the heating/cooling system On or Off.

Probabilistic methods such as Bayesian Networks (BNs) or Conditional Random Fields (CRFs) seem to be suitable for this role. More specifically, BNs could be fed by low-level knowledge (e.g., activities performed by the user, objects recognised in the environment, readings from sensors, and so on) in order to have a perception of the current situation. The capability of BNs to yield the likelihood of a future state enables high-level reasoning and, consequently, allows a DT to predict situations given its previous status. Hence, a BN could be easily compared to the three-tiered SA model described above.

2.4.2.1 Bayesian Networks

Bayesian Networks (BNs) are a computational tool to develop qualitative probabilistic models. BNs are oriented graphs joined to an underlying conditional probability distribution field, where nodes represent random variables and arrows represent causal dependencies that characterize the real system. Models built upon Bayesian Networks can provide valuable decision support for a large range of issues where decision-making under uncertainty is required. They offer the possibility of conducting both deductive and inductive inferences on which scenario analysis, sensitivity analysis and diagnostic assessments can be based. Furthermore, BNs are able to update automatically the conditional probability field when new data are available (De Grassi, Naticchia, Giretti, & Carbonari, 2009).

Formally, Bayesian networks are directed acyclic graphs (DAGs) whose nodes represent variables in the Bayesian sense: they may be observable quantities, latent variables, unknown parameters or hypotheses. Edges represent conditional dependencies; nodes that are not connected (no path connects one node to another) represent variables that are conditionally independent of each other. Each node is associated with a probability function that takes, as input, a particular set of values for the node's parent variables, and gives (as output) the probability (or probability distribution, if applicable) of the variable represented by the node.

Probabilistic inference consists in computing the a posteriori probability of the occurrence of a subset X of events included in the universe U of likely events after observing another subset Y of events of U, typically disconnected from the previous one. Therefore, Bayesian inference consists in calculating the a posteriori probability distribution of a variable following the observation of a group of events. Bayes' theorem is stated mathematically as the following equation:

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)} \tag{2}$$

Where a and b must be different events and $P(b) \neq 0$. $P(a|b)$ is the probability that the event a occurs given that the event b is true. It is called *Conditional Probability* or even *Posterior Probability* of a given b. $P(b|a)$ is also a conditional probability. $P(a)$ and $P(b)$ are the probabilities of observing a and b respectively without any given conditions. They are known as *Marginal probability* or *Prior probability*.

A definition of the Bayesian Networks let $G = (V, E)$ be a directed acyclic graph and let $X = (X_v), v \in V$ be a set of random variables indexed by V. X is a Bayesian network with respect to G if its joint probability density function (with respect to a product measure) can be written as a product of the individual density functions, conditional on their parent variables:

$$p(x) = \prod_{v \in V} p(X_v | X_{pa(v)}) \quad (3)$$

where $pa(v)$ is the set of parents of v (i.e., those vertices pointing directly to v via a single edge). For any set of random variables, the probability of any member of a joint distribution can be calculated from conditional probabilities using the chain rule (given a topological ordering of X) as follows:

$$P(X_1 = x_1, \dots, X_n = x_n) = \prod_{v=1}^{n} P(X_v = x_v | X_{v+1} = x_{v+1}, \dots, X_n = x_n) \quad (4)$$

and using the definition above, this can be written as:

$$P(X_1 = x_1, \dots, X_n = x_n) = \prod_{v=1}^{n} P(X_v = x_v | X_j = x_j \text{ for each } X_j \text{ that is a parent of } X_v) \quad (5)$$

The difference between the expressions *(Eq. 4)* and *(Eq. 5)* is the conditional independence of the variables from any of their non-descendants, given the values of their parent variables.

A basic application of Bayes' theorem is shown in *(Fig. 22)*.

Sick="sick"	Sick="not"
0.1	0.9

Table 1: P(Sick).

Dry="dry"	Dry="not"
0.1	0.9

Table 2: P(Dry).

	Dry="dry"		Dry="not"	
	Sick="sick"	Sick="not"	Sick="sick"	Sick="not"
Loses="yes"	0.95	0.85	0.90	0.02
Loses="no"	0.05	0.15	0.10	0.98

Table 3: P(Loses | Sick, Dry).

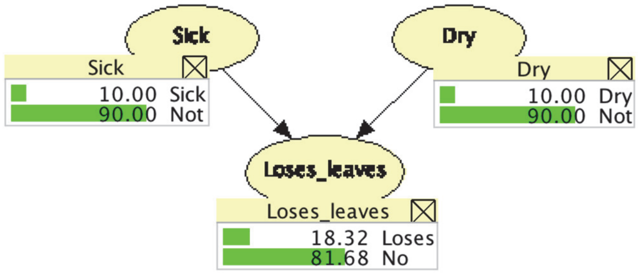


Figure 22, *Example of BN regarding three different events that an apple three may experience. To the left are the tables showing the Marginal probabilities of the events Sick and Dry (10% probability that they occur) and the Conditional probability table (CPT) of the event Loses_leaves. To the right is the network pane where the probabilities of each event are shown* (Hugin).

Bayesian Networks perform three main inference tasks:

- *Inferring unobserved variables*, Bayesian networks are complete models for their variables and their relationships, they can be used to answer probabilistic queries about them. For instance, the network can be used to update knowledge of the state of a subset of variables when other variables (the evidence variables) are observed. This process of computing the posterior distribution of variables given proofs is called *probabilistic inference*. The posterior gives a universal sufficient statistic for detection applications, when choosing values for the variable subset that minimize some expected loss function, for instance the probability of decision error. A Bayesian network can therefore be considered a mechanism for automatically applying Bayes' theorem to complex problems. The most common exact inference methods are: variable elimination, which eliminates (by integration or summation) the non-observed non-query variables one by one by distributing the sum over the product; clique tree propagation, which caches the computation so that many variables can be queried at one time and new evidence can be propagated quickly; and recursive conditioning and AND/OR search, which allow for a space–time trade off and match the efficiency of variable elimination when enough space is used. All of these methods have complexity that is exponential in the network's treewidth. The most common approximate inference algorithms are importance sampling, stochastic MCMC simulation, mini-bucket elimination, loopy belief propagation, generalized belief propagation and variational methods.

- *Parameter learning*, in order to fully specify the Bayesian network and thus fully represent the joint probability distribution, it is necessary to specify for each node *X* the probability distribution for *X* conditional upon *X*'s parents. The distribution of *X* conditional upon its parents may have any form. It is common to work with discrete or Gaussian distributions since that simplifies calculations. Sometimes only constraints on a distribution are known; one can then use the principle of maximum entropy to determine a single distribution, the one with the greatest entropy given the constraints. (Analogously, in the specific context of a dynamic Bayesian network, the conditional distribution for the hidden state's temporal evolution is commonly specified to maximize the entropy rate of the implied stochastic process). Often these conditional distributions include parameters that are unknown and must be estimated from data, e.g., via the maximum likelihood approach. Direct maximization of the likelihood (or of the posterior probability) is often complex given unobserved variables. A classical approach to this problem is the expectation-maximization algorithm, which alternates computing expected values of the unobserved variables conditional on observed data, with maximizing the complete likelihood (or posterior) assuming that previously computed expected values are correct. Under mild regularity conditions this process converges on maximum likelihood (or maximum posterior) values for parameters. A more fully Bayesian approach to parameters is to treat them as additional unobserved variables and to compute a full posterior distribution over all nodes conditional upon observed data, then to integrate out the parameters. This approach can be expensive and lead to large dimension models, making classical parameter-setting approaches more tractable.
- *Structure learning*, in the simplest case, a Bayesian network is specified by an expert and is then used to perform inference. In other applications the task of defining the network is too complex for humans. In this case, the network structure and the parameters of the local distributions must be learned from

data. Automatically learning the graph structure of a Bayesian network (BN) is a challenge pursued within Machine Learning. The basic idea goes back to a recovery algorithm developed by Rebane and Judea Pearl12 and rests on the distinction between the three possible patterns allowed in a 3-node DAG:

Chain: X → Y → Z

Fork: X ← Y → Z

Collider: X → Y ← Z

The first 2 represent the same dependencies X and Z are independent given Y and are, therefore, indistinguishable. The collider, however, can be uniquely identified, since X and Z are marginally independent, and all other pairs are dependent. Thus, while the skeletons (the graphs stripped of arrows) of these three triplets are identical, the directionality of the arrows is partially identifiable. The same distinction applies when X and Z have common parents, except that one must first condition on those parents. Algorithms have been developed to systematically determine the skeleton of the underlying graph and, then, orient all arrows whose directionality is dictated by the conditional independencies observed.

A further feature of Bayesian Networks consists in their flexibility. BNs are powerful knowledge integrators. A Bayesian Network can be built from personal knowledge, systems of equations, historical data series, or any combination of them. Thus, BNs are one of the most reliable and effective tools for solving complex problems systematically currently available.

A particular type of BN is called *Object-Oriented Bayesian Network (OOBN)*. OOBN consists of one or more sub-networks connected through a set of interface nodes. Interface nodes can be input or output nodes. Each sub-network can be a standard Bayesian network or, recursively, an Object-Oriented network, i.e., a Bayesian network formed by sub-networks *(Fig. 23)*. Object Oriented methods were first introduced at the end of the 1980s: the aim was to try to maximize the reuse of single modules (the sub-networks) in the modelling of complex problems, thus controlling the problem of

12 Computer scientist and philosopher, best known for championing the probabilistic approach to artificial intelligence and the development of Bayesian networks. Awarded with the Turing Award in 2011.

the correctness of the implementation and structuring of the overall network. The positive aspects deriving from the OOBN approach reside in the control of complexity, correctness and computational efficiency through a modular approach to modelling. Object Oriented networks therefore find their main reason to exist in the management of complex models. The construction of complex networks requires well-structured procedures for managing information and the development process. In such cases, the knowledge elicitation process will be cyclical, and the network will have to be produced and refined through successive steps. The analyst employs experts to break down the system into its fundamental parts, recognises patterns at different levels (both the organisation of large-scale networks and sub-networks) and must be prepared to update the system as the revision work with the experts proceeds (De Grassi, Naticchia, Giretti, & Carbonari, 2009).

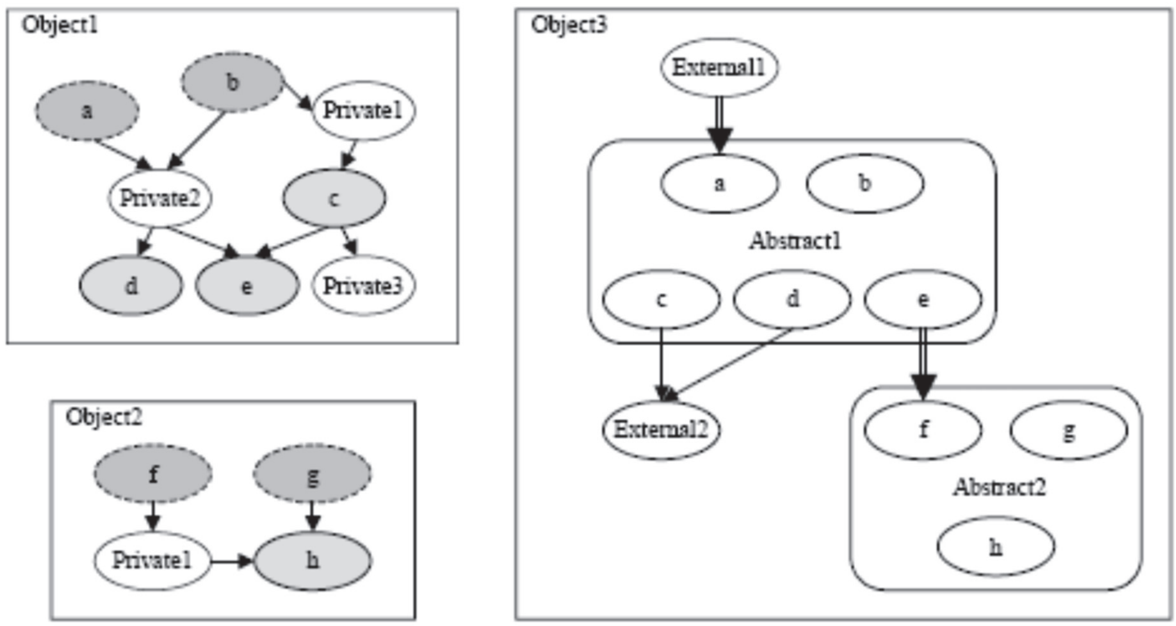


Figure 23, *Example of OOBN. Object3 is a network composed of Object1 and Object2, that are sub-modules. Input nodes are grey with dashed edges. Output nodes are grey with solid edges* (De Grassi, Naticchia, Giretti, & Carbonari, 2009).

2.4.3. Home automation

Home automation is building automation for a home, namely a *Smart Home*. A home automation system will monitor and/or control home attributes such as lighting, climate, entertainment systems, and appliances. It may also include home security such as access control and alarm systems. When connected with the Internet, home devices are an important constituent of the *Internet of Things*. A home automation system typically connects controlled devices to a central hub. The user interface for control of the system uses either wall-mounted terminals, tablet or desktop computers, a mobile phone application, or a Web interface that may also be accessible off-site through the Internet.

Home automation is prevalent in a variety of different realms, including:

- *Heating, ventilation and air conditioning (HVAC)*, it is possible to have remote control of all home energy monitors over the internet incorporating a simple and friendly user interface.
- *Lighting control system*, a network that incorporates communication between various lighting system inputs and outputs, using one or more central computing devices.
- *Occupancy-aware control system*, it is possible to sense the occupancy of the home using smart metres and environmental sensors like CO_2 sensors, which can be integrated into the building automation system to trigger automatic responses for energy efficiency and building comfort applications.
- *Appliance control and integration with the smart grid and a smart metre*, taking advantage, for instance, of high solar panel output in the middle of the day to run washing machines.
- *Home robots and security*, a household security system integrated with a home automation system can provide additional services such as remote surveillance of security cameras over the Internet, or access control and central locking of all perimeter doors and windows.
- *Leak detection, smoke and CO detectors.*
- *Indoor positioning systems (IPS).*
- *Home automation for the elderly and disabled.*

- *Air quality control*, for example, Air Quality Egg is used by people at home to monitor the air quality and pollution level in the city and create a map of the pollution.
- *Smart kitchen*, with refrigerator inventory, premade cooking programs, cooking surveillance, and so on.
- *Voice control devices*, for instance, *Amazon Alexa*13 or *Google Assistant14* used to control home appliances or systems.

Currently, there are several platforms that enable home automation, such as *Home Assistant15*, *openHAB16*, and so on. The interaction with those platforms and users is usually carried out through *flow-based programming tools (FBP)*.

13 http://alexa.amazon.com
14 https://assistant.google.com/
15 https://www.home-assistant.io
16 https://www.openhab.org

Purpose

The aim of this thesis lies in the themes presented so far. Specifically, the need to find reliable solutions for the support of the elderly population for an active and healthy autonomous ageing, and the theme of digital innovation in the field of construction and the built environment, encounter each other and define the basis of this proposal. The objective of this research is therefore the development of Digital Twin models that supporting Ambient Assisted Living. This dissertation, as well as defining a consistent, flexible, and replicable system, focuses on the necessity of a digital innovation in the world of construction nowadays, and therefore on the need to build or renovate buildings not only thinking about *Hardware* components (defined by its tangible physical parts), but also about the *Software* component (defined by a cognitive system capable of addressing the user's needs).

CHAPTER III

METHODOLOGY

In this chapter, we address methods of Digital Twin models development that are suited for our purpose. First, we define the system requirements. Then, we provide the system architecture of our Digital Twin, based on the five-tier system architecture shown in (Lu, et al., 2020). Each layer is then examined in depth in later paragraphs.

3.1 Requirements and functional specifications

Referring to the ageing of the population and the geriatric syndromes which may occur there are currently two main scenarios:

- The elderly are not able to live independently anymore and need to be hosted in care structures. Those scenarios are often refused by older people because they recognize that it would be an irreversible situation. Furthermore, this is the most expensive solution.
- The elderly continue to live at their homes, but need to be assisted by caregivers. This scenario is more appreciated by older people. Nevertheless, they would lose their autonomy as well.

However, both scenarios involve the intervention of assistants. Our system aims at supporting autonomous users in their ADLs. For this purpose, the system must be equipped with an appropriate technological infrastructure that enables the detection of anomalous situations in the home environment. Our prototype is suitable in each of the following contexts:

- New residential buildings.
- Healthcare structures for the elderly.
- Refurbishment or retrofitting of existing dwellings.

In each case, the broad context regards that of the *Smart Environments (Fig. 24)*.

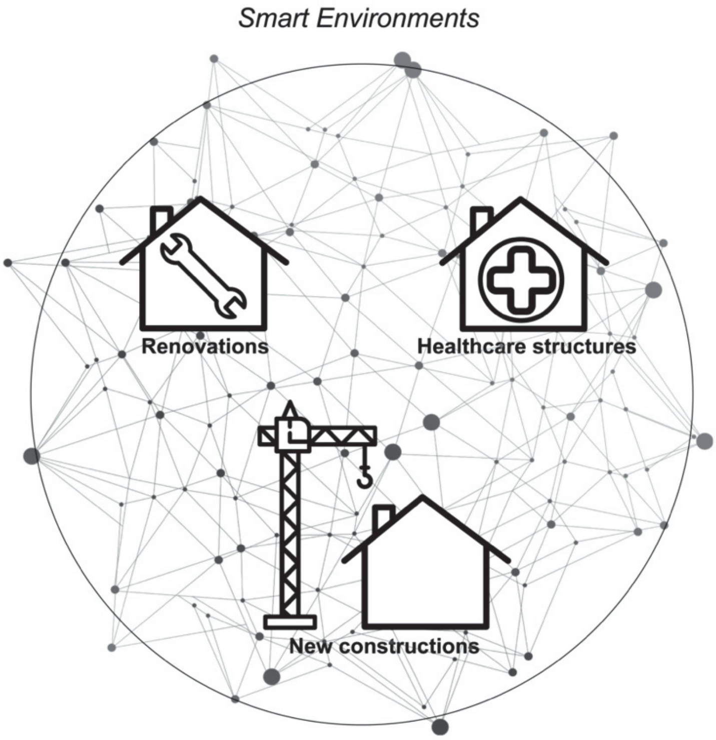


Figure 24, *Context of the prototype.*

The requirements of the prototype are shown below:

- *Flexibility*, even though the prototype has been conceived in an arbitrary scenario, it must be adaptable to different situations. This means that the system could be deployed in other kinds of environments (e.g., other rooms or even buildings) without substantial adjustments.
  Another significant feature in terms of flexibility regards the system capability to adapt to the user. The most appropriate approach to address this demand is to define a system able to learn over time the changing habits of the user and act accordingly.
- *Non-intrusiveness*, one of the main problems of Smart Home systems that imply monitoring the subject is that they are generally not easily accepted by the individual. The awareness of being observed is perceived by him or her as a discomfort. For this reason, it is essential for the system to be as less intrusive as possible and, above all, not perceived by the older person as something extraneous. It is well known that the elderly are attached to their familiar habits and environments, so it is vital not to cause substantial changes to their homes. Particularly, when considering people suffering from cognitive disorders, such

as Alzheimer's, it is a good idea to keep the home environment stable, always putting things in the same place to avoid disorientation.
Furthermore, it is very important not to interfere too much with the user's ADLs: vocal or visual reminders of events or suggestions may be very useful to support the individual in his or her routine, but should not be perceived as invasive. As a matter of fact, one of the most common feelings among people suffering from dementia is the sense of frustration with their condition, which is exacerbated when the patient is constantly being corrected for his/her mistakes.

- *Affordability*, low-cost is a key requirement whether the system is deployed in a private home or in a healthcare structure. To ensure a cost-effective system, it is not enough using inexpensive sensors and devices (generally speaking, the cheapest sensors are binary sensors, i.e., only able to generate an 'on' or 'off' signal), but their distribution must be optimized. For example, in order to understand what activity a person is doing, it may be useful to know how he or she interacts with objects in the house. To do this, it would be sufficient to place a sensor on the object that is able to recognize when it is touched or moved (in the case of a tool) or switched on or off (in the case of an appliance).
It is obvious, however, that it is not possible to provide a sensor for every object in the house, even if it is inexpensive. The choice of sensor arrangement in the living space is therefore particularly important. Devices placed in the appropriate positions guarantee the collection of meaningful data for the recognition of certain activities.

3.2 System Architecture

The system architecture of the proposed DT model has been outlined following the guidelines stated in (Lu, et al., 2020), which defines the structure of a building-level DT *(2.3.1)*. Thus, the model is defined by five layers, namely Data Acquisition Layer, Transmission Layer, Digital Modelling Layer, Data/Model Integration Layer and Service Layer. Our pipeline defines a system able to autonomously perform high-level reasoning to detect anomalies in daily situations and consequently offer support to the user. The architecture of the system is shown in *(Fig. 25)* and each layer is then extensively explained in the following chapters.

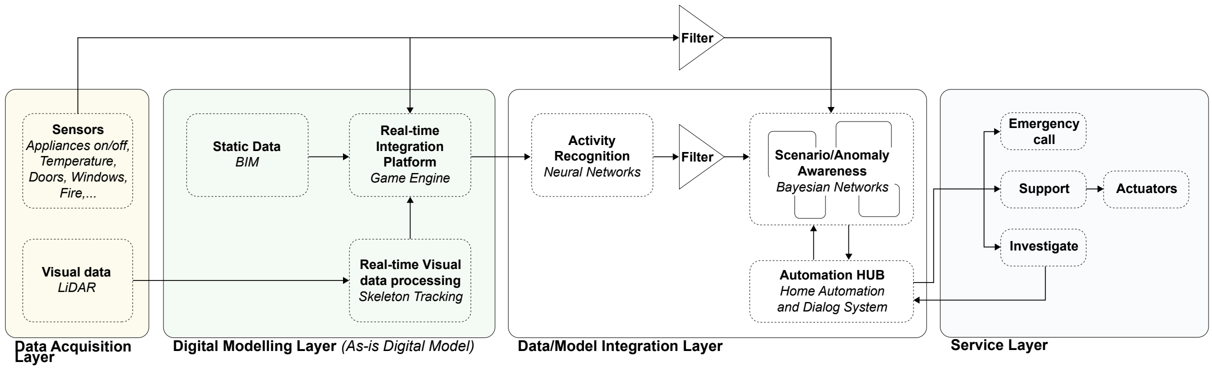


Figure 25, *System architecture of the proposed DT model. The Transmission Layer is represented by arrows, and their directions indicate data flow.*

Overall, our approach has enabled us to define an open and flexible system whose components can also be extended or even replaced in future by more accurate or appropriate intelligent agents. For instance, the Neural Network Unit could be replaced by another similar AI component if necessary. Therefore, the system may also be upgraded later with new technologies.

3.2.1 Data Acquisition Layer

The Data Acquisition Layer is responsible for capturing data. Since the use of visual information (videos and/or images) has become popular and affordable throughout the last years, we have chosen to integrate a 3D camera in our system. Nevertheless, 3D camera alone would not have allowed us to infer situations where environment data are required. For this reason, we have integrated other sensors as well (e.g., on/off detection sensors, temperature sensors, open/close detection sensors, and so forth). This enables a multi-modal system which combines different data typologies to provide a broad vision of what is happening at a time in the environment. Consequently, the use of a multi-modal systems enables higher accuracy.

3.2.1.1 Acquisition of Visual Information

Visual information is necessary to broaden the knowledge of the system about the environment and the user. Visual information could be acquired with various methods that rely on different typologies of devices. Devices that extract 3D information of the environment and its components are:

- *3D camera*
- *LiDAR*

Concerning 3D cameras, they can be further divided in two sub-types, namely Range camera and Stereo camera. The former includes devices that produce 2D images showing the distance to points in a scene from a specific point. The resulting image, the range image, has pixel values that correspond to the distance. If the sensor that is used to produce the range image is properly calibrated the pixel values can be given directly in physical units, such as meters. The latter includes cameras with two or more lenses with separate image sensors or film frame for each lens, which allows the camera to simulate human binocular vision, and therefore capture three-dimensional images.

With regard to LiDAR, it is a method for determining ranges (variable distance) by targeting an object with a laser and measuring the time for the reflected light to return to the receiver. Lidar can also be used to make digital 3-D representations of areas on the earth's surface and ocean bottom, due to differences in laser return times, and by varying laser wavelengths. Lidar uses ultraviolet, visible, or near infrared light to image objects. It can target a wide range of materials, including non-metallic objects, rocks, rain, chemical compounds, aerosols, clouds and even single molecules. A narrow laser beam can map physical features with very high resolutions; for example, an aircraft can map terrain at 30-centimetre resolution or better.

Since 3D information regarding the environment are extremely sizable for our purpose, we have integrated a LiDAR camera in our system. The used device is the *Intel RealSense LiDAR Camera L515*[17]. It is introduced as a revolutionary solid-state

17 https://www.intelrealsense.com/lidar-camera-l515

LiDAR depth technology designed for indoor applications. Furthermore, this device is suitable for applications that require depth data at high resolution and high accuracy. The dimensions of the L515 camera enable the *non-intrusiveness* requirement to be met.

3.2.1.2 Other sensors

In Smart Homes, sensors are usually deployed to collect several data concerning building's elements, environmental aspects, appliances, services and so on. Those data are extremely sizable for Digital Twin models. In Ambient Assisted Living, sensors that extract data from appliances or specific building's elements such as doors and windows could be useful to detect anomalies while the user is performing ADLs. Although no scene has been set up for our research, we have considered a list of sensors, that could be easily deployed in a Smart Home and can collect valuable data to feed the High-level knowledge reasoner. Sensors considered in our prototype are:

- *Water flow switch*, Is a mechanical switch that is turned on or off depending on the flow of a fluid (water, air, and so on). The switch typically operates through a paddle which gets displaced due to the energy of the fluid moving past it. This sensor is a special electric device with a very simple design and of a small size (Water Flow Sensor, s.d.).
- *Open-closed sensors*, They can be used for opened/closed applications that provide information on the status of doors and windows. It could be used to know if the fridge is closed or open as well.
- *Temperature sensors*, These sensors can be even plugged to household appliances such as cooktop, oven and so on. In those cases, they monitor cooking temperature, glass surface temperature, and allow cooking control. Nevertheless, temperature sensors could be used to monitor indoor and outdoor temperature.

3.2.2 Transmission Layer

The Transmission Layer, represented by the solid arrows in the figure, allows the transfer of data within the system. It is based on a specific data-interchange format, namely *JSON (JavaScript Object Notation)*. JSON is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. JSON is built on two structures:

1. A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
2. An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures (json.org, s.d.).

3.2.3 Digital Modelling Layer

The Digital Modelling Layer is based on a game engine, namely *Unity*, which allows the creation of an *As-is Digital Model*. The model is the virtual illustration of the real asset and consists in a real-time representation of the environment and its user. It is built upon two types of data, i.e., *static data* and *dynamic data*. The former comes from Building Information Modelling (BIM) and gives both graphic and semantic representation of the building and its parts. Since those data do not change in a short period of time, they are considered static. The latter regards the real-time information which comes from the Data Acquisition Layer. The output of the 3D camera feeds an AI algorithm responsible for Human Skeleton Tracking (Nuitrack SDK algorithm). This tool detects the human skeleton and articulates it into joints and bones. The real-time coordinates of the joints are its output, which is fed directly into the game engine as well as the sensor readings, to dynamically update the model.

3.2.3.1 Game engine

A game engine is the core software necessary for a game program to properly run or to develop other interactive contents such as architectural 3D visualizations and real-time animations. Developers can use game engines to construct contents or video game consoles and other types of computers. The main functionality typically provided by a game engine may include a rendering engine for 2D or 3D graphics, a physics engine or collision detection, sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, scene graph, and video support for cinematics18.

The features of game engines define a convenient environment not only for game developers, but also to develop Digital Twins. As a matter of fact, since a DT is the virtual counterpart of a real asset, such platforms are suitable to develop cyber-physical systems. Above all, a programmable environment that copes with geometry, physics and agents represents the ideal platform to build those systems.

The decision to use the Unity19 game engine platform among the many available is due to the fact that it is beneficial in our context. So far, virtual simulation platforms such as Anylogic20 have been employed in the built environment for similar purposes. Those programs support the three main simulation paradigms: discrete events (DE), system dynamics (SD) and agent-based (AB). Though, a game engine has more features and, therefore, potentialities than a simulation tool. Furthermore, other available game engines are more focused on the release of games rather than interactive contents in other fields, whereas Unity is not. However, a similar platform (Unreal Engine)21 has been taken into account. Unreal Engine is one of the most open and advanced real-time 3D creation tools as well as Unity. Both tools are capable of

18 https://en.wikipedia.org/wiki/Game_engine
19 https://unity.com
20 https://www.anylogic.com/
21 https://www.unrealengine.com

producing AAA-quality graphics and have great bridges between most of the industry-standard software. They both provide an extensive toolbox including terrain editor, animation, physics simulation, VR Support, and so on. Although both systems offer great potential, for our purpose Unity has been preferred for the following reasons:

- *Popularity*, Unity is more used in the scientific field than Unreal.
- *Language*, Unity uses C# while Unreal uses C++. Hence, Unity is preferable due to the faster and easier nature of the C# coding.
- *Documentation*, both offer worthy and detailed documentation explaining their tools and features. However, there is a wider range of Unity courses.
- *Asset Store*, Unity has a wide range of mods in comparison to Unreal. The latter has around 10000 assets while the former has 31000 assets.
- *Learning Curve*, Unity is much easier to learn than Unreal due to its intuitive interface.

For all these reasons, Unity has been employed as the platform where the Digital Modelling Layer is rooted and implemented in real-time. In our case, the game engine is fed by skeleton real-time data (dynamic data) and BIM data (static data) to define a virtual copy of the real asset which takes into account both the user and the environment.

3.2.3.2 Skeleton Tracking

Since the model aims at recognizing activities undertaken by the user to infer anomalies, a Skeleton Tracking system is required. Skeleton Tracking is the processing of 3D or depth image data to establish the position of various human skeleton joints. For example, skeleton tracking determines where a user's head, hands and centre of mass are. It provides x, y and z values for each of these skeleton points. Skeleton Tracking algorithms are complex AI models that use matrix transforms, Machine Learning, and other means to calculate the skeleton joint coordinates (Webb & Ashley, 2012). Those algorithms could be integrated in game engines.

Skeleton Tracking systems may offer different skeleton structures. This means that the shape of the skeleton, the number of joints, and their position could vary depending on the system taken into account *(Fig. 26)*. For example, the Kinect v2 skeleton (which has been used to build the NTU-RGBD dataset) has 25 joints as well as the OpenPose skeleton, while the Nuitrack skeleton has 19 joints. Even though Kinect and OpenPose have the same number of joints, their positions is different: Kinect v2 has only one joint on the head, whereas OpenPose has five joints on the head.

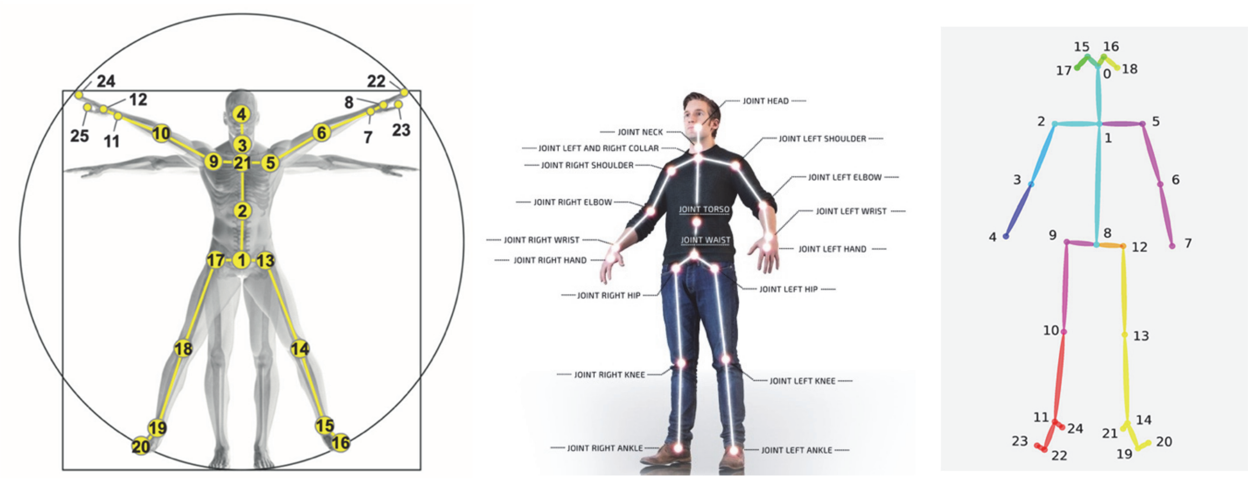


Figure 26, *From left to right: Kinect v2 skeleton, Nuitrack skeleton, OpenPose skeleton.*

Among all the Skeleton Tracking systems available, we have employed the cross-platform Nuitrack AI in our system due to several reasons. Firstly, in spite of the fact that the Kinect v2 is one of the most popular systems (it has been used in several papers), it has been put out of production. Microsoft has indeed replaced this system with the Azure Kinect22. This new AI-powered kit is a powerful tool that exploits the Azure platform. Nevertheless, its skeleton has 32 joints, and its shape is definitely different compared to that of the most common skeleton dataset such as NTU-RGBD (that has been produced using a Kinect v2). Secondly, the OpenPose system relies on 2D images instead of 3D information. Thus, it would only enable the NNs to be fed with joints data, but would not allow the creation of a 3D real-time representation into the game engine.

3.2.3.3 Building Information Modelling

22 https://azure.microsoft.com/it-it/services/kinect-dk

In an environment where changes are frequent (the user real-time situation may change each frame), the static data are information regarding anything that cannot change (or be changed) in the time of an action. This kind of data in a Digital Twin for Ambient Assisted Living should concern the building and its elements. Since BIM is a container of building information, a BIM model of an arbitrary Smart Home for elderly people *(Fig. 27)* has been created and integrated in our as-is model.

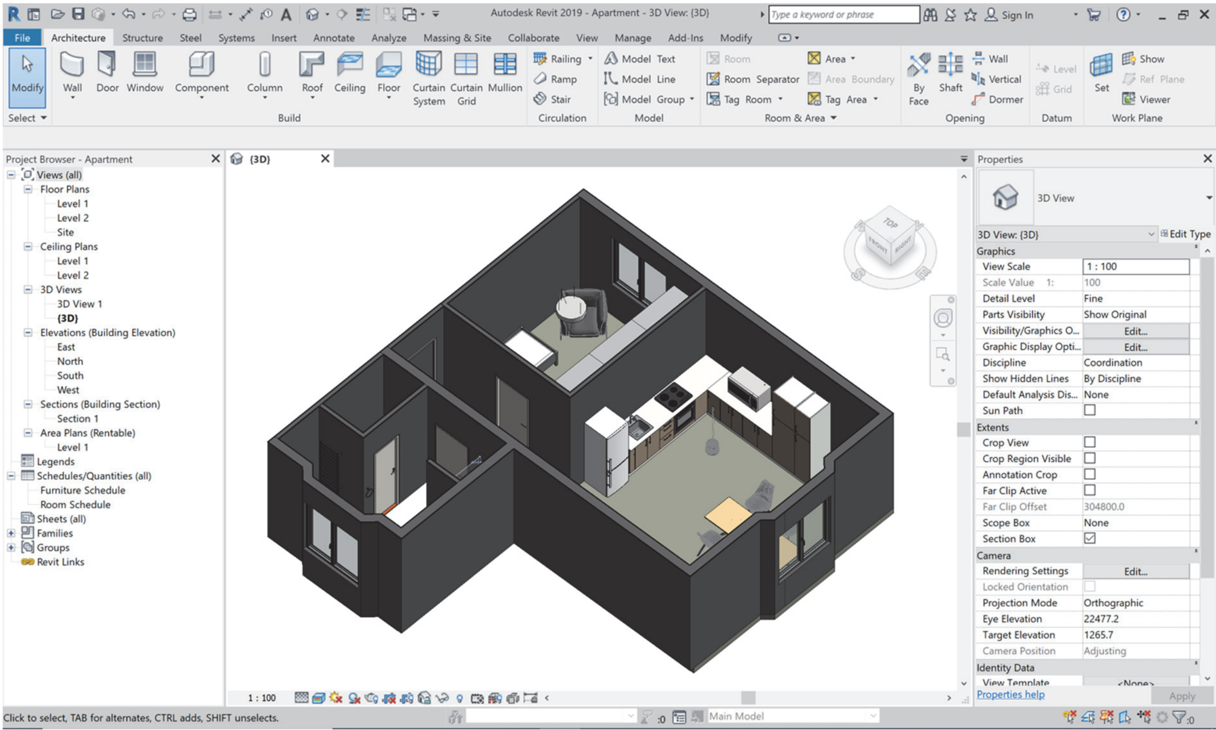


Figure 27, *BIM model of an arbitrary scenario (created in Revit).*

3.2.4 Data/Model Integration Layer

This layer defines the intelligent tier of the whole system that allows the building to actively participate in the user's life. In our case, namely an ambient Assisted Living Scenario, the building should be able to support the inhabitant through an automatic knowledge system that makes it responsive and proactive to everyday situations. The Digital Twin of a building represents the automatic knowledge system required and, consequently, facilitates the creation of a cognitive building. The Data/Model Integration Layer of the DT is therefore responsible for analysing and processing the data captured and fixed by the previous layers. Hence, its accuracy influences the effectiveness of the whole system. Since our prototype aims at supporting the elderly in his or her home environment through the detection of anomalies in daily activities, it requires intelligent agents for:

- Action/activity detection
- Reasoning on the current situation

These two required tasks have led us to define a system that works at two different level of knowledge *(Fig. 28)*:

1. *Low-level*, responsible for action/activity detection.
2. *High-level*, responsible for reasoning at higher levels to recognize anomalies.

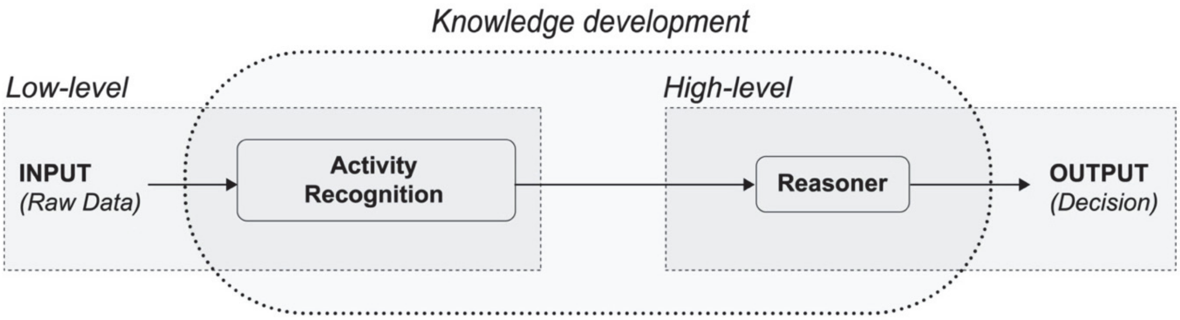


Figure 28, *Knowledge development methodology, from raw data to high-level decision.*

The knowledge development methodology proposed above is based on a knowledge hierarchy *(Fig. 29). It* offers a model that structures data, information, and knowledge in different levels and defines procedures that transform each of these entities into another entity at a higher level (Barb & Shyu, 2008). Data are the input and are viewed as atomic, discrete properties of an object. Through collection and quantification, we derive information by giving meaning to similar data entities. In an image database, low-level features extracted from images constitute information that can be used to further derive knowledge. Knowledge is viewed as information assimilated in a personal and subjective manner that can be used to make decisions. It can be derived either by instruction or experience. Knowledge is further used to gain wisdom, which is an exclusive human feature that helps us integrate tacit knowledge into decision-making processes.

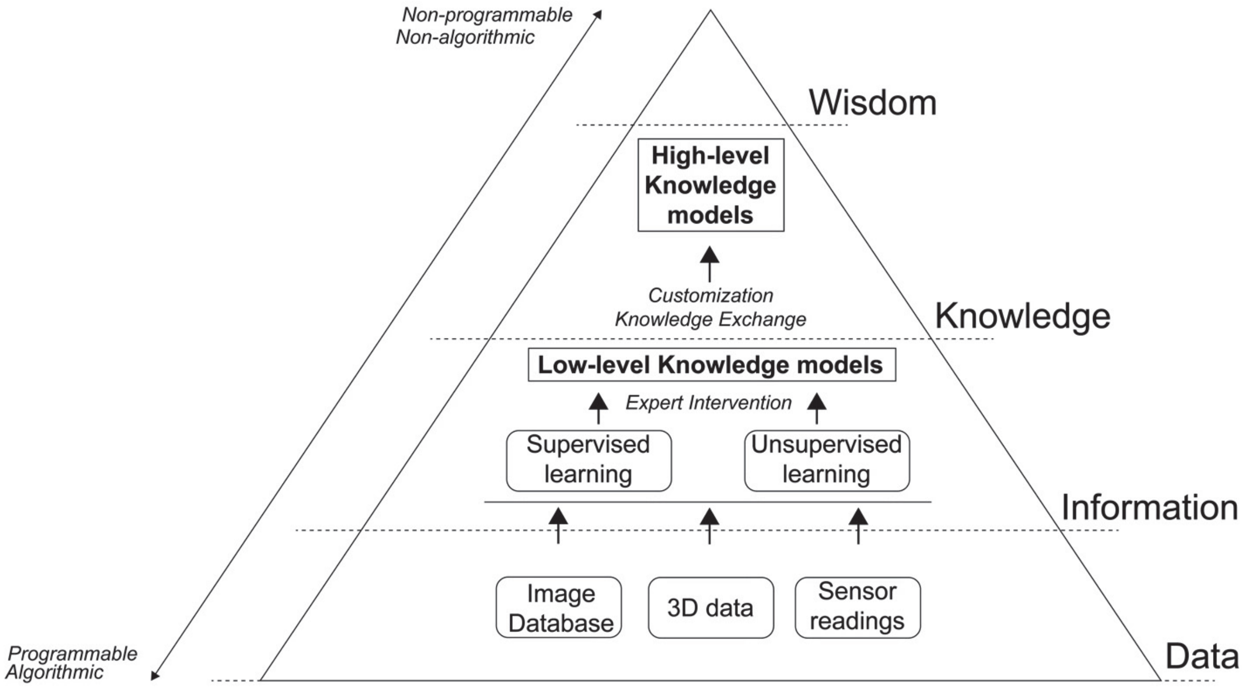


Figure 29, *Knowledge pyramid adapted and applied to the computer-based knowledge representation* (Barb & Shyu, 2008).

To develop high-level knowledge, our Data/Model Integration Layer consists of three main AI sub-systems: Neural Networks, Bayesian Networks and a Dialog System. These tools are the core of the DT model. They perform real-time reasoning and define a sequential system that is initially fed with the coordinates of the human skeleton joints (data). The Neural Networks (Unsupervised learning model) process these raw data to yield the probability that an action has been undertaken by the user. The output of the NNs is still a low-level knowledge of what is happening in the scenario. The actions and the sensor readings are thus managed by a high-level reasoner, i.e., the Bayesian Networks, just after being filtered to reduce noise. If required, the BNs send a message to the Automation HUB that offers direct support to the user or investigates thoroughly in order to make better decisions. The further investigation is carried out through the Dialog System.

The data transfer within NNs and BNs, and BNs and the Automation HUB have not been investigated thoroughly since they are merely technical issues. The former could be solved by interposing a filter between the systems as indicated in the system architecture *(Fig. 25)*. On the one hand, the Bayesian Network requires discrete inputs due to its configuration. On the other hand, the output of the Neural Networks is continuous and, therefore, it may lead to ambiguous circumstances if directly fed into the BNs. An algorithm able to filter the actions detected by the NNs should be embedded in the pipeline. Furthermore, even the raw sensor readings that are directed to the BNs should be pre-processed in order to avoid noise as well. These upgrades of the system as well as the integration with the Automation HUB could be developed using the API of the Bayesian Network software.

3.2.4.1 Low-level Knowledge Model

The first level is responsible for action/activity detection. The term *low-level knowledge* comprehends both the type of system chosen for the task of recognizing actions undertaken by the user and the quality of the respective input and output data. If we are talking specifically about the action recognition model, we could define it as a low-level system since it is not able to perform contextualized reasoning on the current situation of the environment. The knowledge regarding which action or activity the inhabitant is pursuing at a time during the day can be achieved through the use of implicit methods. Since how this task is executed is not relevant, and there is an extensive sampling of data concerning the recognition of actions and activities performed by individuals, it is reasonable to use implicit models such as Neural Networks. Those kinds of models are not accessible because the operations they perform are hidden and do not allow interaction. NNs are implicit models that are trained on Big Data (pre-collected datasets) to carry out results. Many NNs models are available online and are suitable for our purpose *(see Tab. IV - chapter 2.4.1.1)*.

Nevertheless, it is essential to make a distinction between two types of action that the user might undertake:

- actions that produce a visible effect when they are completed. Examples of such actions could be putting a shoe, a jacket, a cap on.
- actions that do not produce a visible effect when they are completed. For instance, drink water, jumping, drop something and so on.

While the first kind of action might be also detected using NNs that recognizes objects (the presence of a shoe or a jacket or a cap would imply that one of the actions listed above has been executed), the second kind of action is more difficult to classify without training the NNs on activities instead of objects. Since NNs for action/activity recognition reveal both typologies, we have employed this system in our prototype to define a complete system.

A huge number of NNs models for activity recognition task are emerging due to their effectiveness. As explained in chapter 2.4.1.1, CNNs have been increasingly exploited for this task in the last years. Visual-information activity recognition has demonstrated to be one of the most accurate systems. However, GCN models – a generalization of CNNs - have emerged, and their potential and accuracy has been proved. GCNs usually rely on skeleton information such as the coordinates of the skeleton joints, expressed by graphs. This property allows us to build a light-weight system. As a matter of fact, while CNNs rely on RGB images or videos (the size of image or video datasets ranges from around 100 GB up to 1 TB), GCNs are trained on 3D skeleton datasets (in this case, the size ranges from 5 GB to 10 GB). Thus, GCNs allow a faster training phase as well.

Among all the models reviewed *(see Tab. IV - chapter 2.4.1.1)*, we have integrated in our pipeline the MS-G3D[23] system for the following reasons:

- works on 3D data
- Works with GCNs exploiting skeleton information (light-weight system)
- Outperforms existing methods for action recognition by a sizable margin on three large-scale datasets: *NTU RGB+D 60*, *NTU RGB+D 120*, *Kinetics Skeleton 400*.
- Pre-trained models are available online
- Programmed in python language

The MS-G3D model is based on the Spatial-Temporal Graph Convolutional Network (ST-GCN) proposed by (Yan, Xiong, & Lin, 2018) and the Two-Stream Adaptive Graph

23 A powerful feature extractor based on Graph Convolutional Networks: https://github.com/kenziyuliu/MS-G3D.

Convolutional Network (2s-AGCN) proposed by (Shi, Yifan, Cheng, & Lu, 2019). A detailed explanation of how this model works follows in the next chapter.

3.2.4.2 MS-G3D

The following description is the one provided by the developers of the model under consideration (Liu, Zhang, Chen, Wang, & Ouyang, 2020).

Notations. A human skeleton graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the set of N nodes representing joints, and $\mathcal{E}$ is the edge set representing bones captured by an adjacency matrix $A \in \mathbb{R}^{N\times N}$ where initially $A_{i,j} = 1$ if an edge directs from v_1 to v_j and 0 otherwise. A is symmetric since $\mathcal{G}$ is undirected. Actions as *graph sequences* have a node features set $\mathcal{X} = \{x_{t,n} \in \mathbb{R}^C \mid t, n \in \mathbb{Z}, 1 \leq t \leq T, 1 \leq n \leq N\}$ represented as a feature tensor $\mathbf{X} \in \mathbb{R}^{T\times N\times C}$, where $x_{t,n} = \mathbf{X}_{t,n,:}$ is the C dimensional feature vector for node v_n at time t over a total of T frames. The input action is thus adequately described by $\mathbf{A}$ structurally and by $\mathbf{X}$ feature-wise, with $\mathbf{X}_{t,n}$ $\in \mathbb{R}^{N\times C}$ being the node features at time t. $\Theta^{(l)} \in \mathbb{R}^{C_l\times C_{l+1}}$ denotes a learnable weight matrix at layer l of a network.

Graph Convolutional Nets (GCNs). On skeleton inputs defined by features $\mathbf{X}$ and graph structure $\mathbf{A}$, the layer-wise update rule of GCNs can be applied to features at time t as:

$$\mathbf{X}_t^{(l+1)} = \sigma\left(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}_t^{(l)}\Theta^{(l)} \right) \tag{6}$$

where $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the skeleton graph with added *self-loops* to keep identity features, $\widetilde{\mathbf{D}}$ is the diagonal degree matrix of $\widetilde{\mathbf{A}}$, and $\sigma(\cdot)$ is an activation function. The term $\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}_t^{(l)}$ can be intuitively interpreted as an approximate spatial mean feature aggregation from the direct neighbourhood followed by an activated linear layer.

Biased Weighting Problem. Under the spatial aggregation framework in *(Eq. 6)*, existing approaches employ *higher-order polynomials* of the adjacency matrix to aggregate multi-scale structural information at time t, as:

$$\mathbf{X}_t^{(l+1)} = \sigma\,(\sum_{k=0}^{K} \widehat{\mathbf{A}}^k\, \mathbf{X}_t^{(l)} \Theta_{(k)}^{(l)}) \tag{7}$$

where K controls the number of scales to aggregate. Here, $\widehat{\mathbf{A}}$ is a normalized form of $\mathbf{A}$. More in general, one can use $\widehat{\mathbf{A}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}$ from GCNs. It is easy to see that $\mathbf{A}_{i,j}^k = \mathbf{A}_{j,i}^k$ gives the number of length k walks between v_i and v_j, and thus the term $\widehat{\mathbf{A}}^k\mathbf{X}_t^{(l)}$ is performing a *weighted* feature average based on the number of such walks. However, it is clear that there are drastically more possible length k walks to closer nodes than to the actual k-hop neighbours due to cyclic walks. This causes a bias towards the local region as well as nodes with higher degrees. The node self-loops in GCNs allow even more possible cycles (as walks can always cycle on self-loops) and thus amplify the bias. Under multi-scale aggregation on skeleton graphs, the aggregated features will thus be dominated by signals from local body parts, making it ineffective to capture long-range joint dependencies with higher polynomial orders.

Disentangling Neighbourhoods. To address the above problem, they first define the k-adjacency matrix $\widetilde{\mathbf{A}}_{(k)}$ as:

$$[\widetilde{\mathbf{A}}_{(k)}]_{i,j} = \begin{cases} 1 & \text{if } d(v_i, v_j\,) = k, \\ 1 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $d(v_i, v_j\,)$ gives the shortest distance in number of hops between v_i and v_j. $\widetilde{\mathbf{A}}_{(k)}$ is thus a generalization of $\widetilde{\mathbf{A}}$ to further neighbourhoods, with $\widetilde{\mathbf{A}}_{(1)} = \widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{A}}_{(0)} = \mathbf{I}$. Under spatial aggregation in *(Eq. 9)* the inclusion of self-loops in $\widetilde{\mathbf{A}}_{(k)}$ is critical for learning the *relationships* between the current joint and its k-hop neighbours, as well as for keeping each joint's *identity* information when no k-hop neighbours are available. Given that N is small, $\widetilde{\mathbf{A}}_{(k)}$ can be easily computed, e.g., using *differences* of graph powers as $\widetilde{\mathbf{A}}_{(k)} = \mathbf{I} + \mathbf{1}\big(\widetilde{\mathbf{A}}^k \geq 1\big) - \mathbf{1}\big(\widetilde{\mathbf{A}}^{k-1} \geq 1\big)$. Substituting $\widehat{\mathbf{A}}^k$ with $\widetilde{\mathbf{A}}_{(k)}$ in *(Eq. 7)*, they arrive at:

$$\mathbf{X}_t^{(l+1)} = \sigma\,(\sum_{k=0}^{K} \widetilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}}\, \widetilde{\mathbf{A}}_{(k)} \widetilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}} \mathbf{X}_t^{(l)} \Theta_{(k)}^{(l)}) \tag{9}$$

where $\tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}}\tilde{\mathbf{A}}_{(k)}\tilde{\mathbf{D}}_{(k)}^{-\frac{1}{2}}$ is the normalized k-adjacency. Unlike the previous case where possible length k walks are predominantly conditioned on length $k-1$ walks, the proposed disentangled formulation in *(Eq. 9)* addresses the biased weighting problem by removing redundant dependencies of distant neighbourhoods' weighting on closer neighbourhoods. Additional scales with larger k are therefore aggregated in an *additive* manner under a multi-scale operator, making long-range modeling with large values of k to remain effective. The resulting k -adjacency matrices are also sparser than their exponentiated counterparts, allowing more efficient representations.

G3D: Unified Spatial-Temporal Modelling. Most existing work treats skeleton actions as a sequence of disjoint graphs where features are extracted through spatial-only (e.g., GCNs) and temporal-only (e.g., TCNs) modules. They argue that such factorized formulation is less effective for capturing complex spatial-temporal joint relationships. Clearly, if a strong connection exists between a pair of nodes, then during layer-wise propagation the pair should incorporate a significant portion each other's features to reflect such a connection. However, as signals are propagated across spacetime through a series of local aggregators (GCNs and TCNs alike), they are weakened as redundant information is aggregated from an increasingly larger spatial-temporal receptive field. The problem is more evident if one observes that GCNs do not perform a weighted aggregation to distinguish each neighbour.

Cross-Spacetime Skip Connections. To tackle the above problem, we propose a more reasonable approach to allow cross-spacetime *skip connections*, which are readily modelled with cross-spacetime *edges* in a spatial-temporal graph. Firstly, consider a sliding temporal window of size τ over the input graph sequence, which, at each step, obtains a spatial-temporal subgraph $\mathcal{G}_{(\tau)} = (\mathcal{V}_{(\tau)}, \mathcal{E}_{(\tau)})$ where $\mathcal{V}_{(\tau)} = \{\mathcal{V}_1 \cup \ldots \cup \mathcal{V}_\tau\}$ is the union of all node sets across τ frames in the window. The initial edge set $\mathcal{E}_{(\tau)}$ is defined by tiling $\tilde{\mathbf{A}}$ into a block adjacency matrix $\tilde{\mathbf{A}}_{(\tau)}$, where:

$$\tilde{\mathbf{A}}_{(\tau)} = \begin{bmatrix} \tilde{\mathbf{A}} & \cdots & \tilde{\mathbf{A}} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{A}} & \cdots & \tilde{\mathbf{A}} \end{bmatrix} \in \mathbb{R}^{\tau N \times \tau N} \tag{10}$$

Intuitively, each submatrix $[\widetilde{\mathbf{A}}_{(\tau)}]_{i,j} = \widetilde{\mathbf{A}}$ means every node in $\mathcal{V}_i$ is connected to itself and its 1-hop spatial neighbours at frame j by *extrapolating* the frame-wise spatial connectivity (which is $[\widetilde{\mathbf{A}}_{(\tau)}]_{i,j}$ for all i) to the temporal domain. Thus, each node within $\mathcal{G}_{(\tau)}$ is densely connected to itself and its 1-hop spatial neighbours across all τ frames. We can easily obtain $\mathbf{X}_{(\tau)} \in \mathbb{R}^{T\times N\tau\times C}$ using the same sliding window over $\mathbf{X}$ with zero padding to construct T windows. Using *(Eq. 6)*, they thus arrive at a unified spatial-temporal graph convolutional operator for the t^{th} temporal window:

$$[\mathbf{X}_{(\tau)}^{(l+1)}]_t = \sigma\,(\widetilde{\mathbf{D}}_{(\tau)}^{-\frac{1}{2}}\widetilde{\mathbf{A}}_{(\tau)}\widetilde{\mathbf{D}}_{(\tau)}^{-\frac{1}{2}}[\mathbf{X}_{(\tau)}^{(l)}]_t\Theta^{(l)} \tag{11}$$

Dilated Windows. Another significant aspect of the above window construction is that the frames need not to be adjacent. A *dilated* window with τ frames and a dilation rate d can be constructed by picking a frame every d frames, and reusing the same spatial-temporal structure $\widetilde{\mathbf{A}}_{(\tau)}$. Similarly, they can obtain node features $\mathbf{X}_{(\tau,d)} \in \mathbb{R}^{T\times N\tau\times C}$ ($d = 1$ if omitted) and perform layer-wise update as in *(Eq. 11)*. Dilated windows allow larger temporal receptive fields *without* growing the size $\widetilde{\mathbf{A}}_{(\tau)}$, analogous to how dilated convolutions keep constant complexities.

Multi-Scale G3D. It can also be possible to integrate the proposed disentangled multi-scale aggregation scheme *(Eq. 9)* into G3D for multi-scale reasoning *directly* in the spatial-temporal domain. They derive the MS-G3D module from *(Eq. 11)* as:

$$[\mathbf{X}_{(\tau)}^{(l+1)}]_t = \sigma\,(\sum_{k=0}^{K}\widetilde{\mathbf{D}}_{(\tau,k)}^{-\frac{1}{2}}\widetilde{\mathbf{A}}_{(\tau,k)}\widetilde{\mathbf{D}}_{(\tau,k)}^{-\frac{1}{2}}[\mathbf{X}_{(\tau)}^{(l)}]_t\Theta_{(k)}^{(l)}) \tag{12}$$

where $\widetilde{\mathbf{A}}_{(\tau,k)}$ and $\widetilde{\mathbf{D}}_{(\tau,k)}$ are defined similarly as $\widetilde{\mathbf{A}}_{(k)}$ and $\widetilde{\mathbf{D}}_{(k)}$ respectively. Remarkably, their proposed disentangled aggregation scheme complements this unified operator, as G3D's increased node degrees from spatial-temporal connectivity can contribute to the biased weighting problem.

Overall Architecture. The final model architecture is illustrated in *(Fig. 30)*. On a high level, it contains a stack of r spatial-temporal graph convolutional (STGC) blocks to extract features from skeleton sequences, followed by a global average pooling layer and a softmax classifier. Each STGC block deploys two types of pathways to simultaneously capture complex regional spatial-temporal joint correlations as well as long-range spatial and temporal dependencies:

1. The G3D pathway first constructs spatial-temporal windows, performs disentangled multi-scale graph convolutions on them, and then *collapses* them with a fully connected layer for window feature readout. The extra dotted G3D pathway indicates the model can learn from multiple spatial-temporal contexts concurrently with different τ and d.
2. The factorized pathway augments the G3D pathway with long-range, spatial-only, and temporal-only modules: the first layer is a multi-scale graph convolutional layer capable of modelling the entire skeleton graph with the maximum K; it is then followed by two multi-scale temporal convolutions layers to capture extended temporal contexts. The outputs from all pathways are aggregated as the STGC block output, which has 96, 192, and 384 feature channels respectively within a typical $r = 3$ block architecture. Batch normalization and ReLU is added at the end of each layer except for the last layer. All STGC blocks, except the first, downsample the temporal dimension with stride 2 temporal conv and sliding windows.

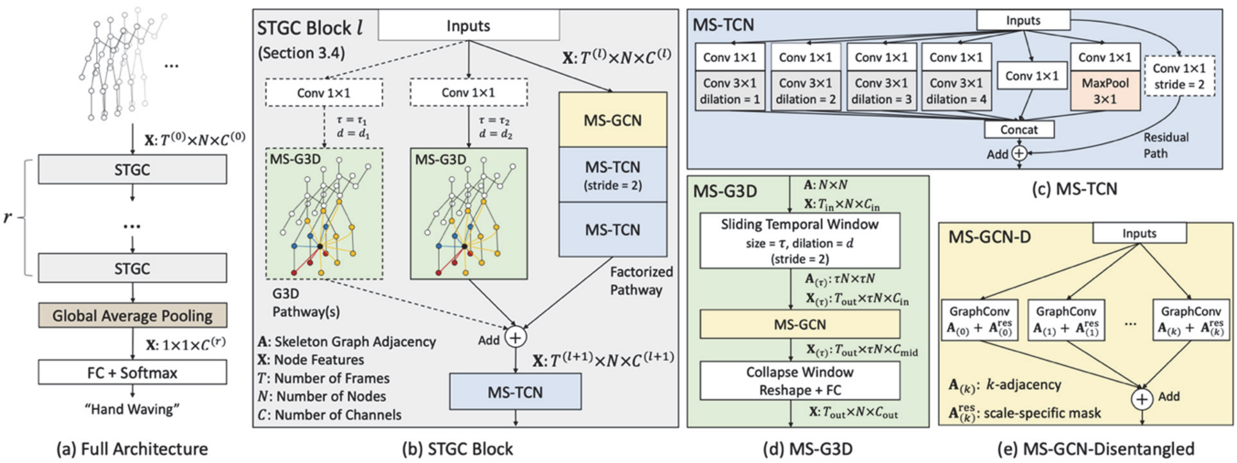


Figure 30, *"TCN", "GCN", prefix "MS-", and suffix "-D" denotes temporal and graph convolutional blocks, and multi-scale and disentangled aggregation. Each of the* r *STGC*

blocks (b) deploys a multi-pathway design to capture long-range and regional spatial-temporal dependencies simultaneously. Dotted modules, including extra G3D pathway, 1×1 conv, and strided temporal convolutions, are situational for model performance/complexity trade-off (Liu, Zhang, Chen, Wang, & Ouyang, 2020).

Multi-Scale Temporal Modelling. The spatial-temporal windows $\mathcal{G}_{(\tau)}$ used by G3D are a closed structure by themselves, which means G3D must be accompanied by temporal modules for cross-window information exchange. Many existing works perform temporal modelling using temporal convolutions with a fixed kernel size $k_t \times 1$ throughout the architecture. As a natural extension to our multi-scale spatial aggregation, they enhance vanilla temporal convolutional layers with multi-scale learning. To lower the computational costs due to the extra branches, they deploy a bottleneck design, fix kernel sizes at 3×1, and use different dilation rates instead of larger kernels for larger receptive fields. They also use residual connections to facilitate training.

Adaptive Graphs. To improve the flexibility of graph convolutional layers which performs homogeneous neighbourhood averaging, they add a simple learnable, unconstrained graph residual mask $\mathbf{A}^{res}$ to every $\widetilde{\mathbf{A}}_{(k)}$ and $\widetilde{\mathbf{A}}_{(\tau,k)}$ to strengthen, weaken, add, or remove edges dynamically.

Joint-Bone Two-Stream Fusion. Inspired by the two-stream methods in (Shi, Yifan, Cheng, & Lu, 2019) and the intuition that visualizing *bones* along with joints can help humans recognize skeleton actions, they use a two-stream framework where a separate model with identical architecture is trained using the bone features initialized as vector differences of adjacent joints directed away from the body centre. The softmax scores from the joint/bone models are summed to obtain final prediction scores. Since skeleton graphs are *trees*, we add a zero bone vector at the body centre to obtain N bones from N joints and reuse $\mathbf{A}$ for connectivity definition.

3.2.4.3 High-level Knowledge Model

The second level is responsible for reasoning at a higher level on data that have been pre-processed by the low-level knowledge model. The term *high-level* knowledge comprehends both the system chosen for the task of reasoning and the quality of its output. A *high-level Reasoner* built upon explicit domain knowledge is suitable for this role. The requirements of the high-level reasoner are:

- *Accessibility*, the model and its parts have to be open and accessible. Implicit models such as NN would be not convenient for this role.
- *Scalability*, as the DT prototype is built in an arbitrary scenario, some conditions and constraints may vary in either different homes or environments. Furthermore, our system still has limited applications, whereas situations where support is required in a real environment would be many more in number. To solve this issue, the high-level Reasoner model must handle scalability.
- *Reusability*, although situations (homes, environments and users) may be different and may change depending on numerous variables, AAL scenarios all have similar aspects. Therefore, the model must have the capability to adapt to other scenarios without dramatic modifications.
- *Non-intrusiveness*, it is crucial not to interfere too much with the user's ADLs. The system should be able to support the individual only if necessary.

This task could be carried out by at least two different methods:

1. Rule-based model
2. Probabilistic model

The first approach uses rule-based algorithms that could be easily written depending on the services needed. Rules are usually *if, else* conditions. A rule-based reasoner has been proposed in (De Paola, et al., An ambient intelligence system for assisted living, 2017). Their module is fed with the output of an activity recognition module (in that case, activity recognition is performed through a Bayesian Network), and takes simple decisions such as turning the heating/cooling system On or Off, and recognizes some health anomalies. For instance, their module that identifies the user's health anomalous conditions is:

anomaly_high_blood_pressure(User, UserCondition) :-
    bloodPressure(high),

activity(User, lying_on_bed),

UserCondition = anomaly.

Their module responsible for turning on the air conditioner if necessary is:

turn_on_air_conditioner(User, Room, RoomTemp) :-

user_unsatisfied(User),

too_cold(User, RoomTemp),

air_conditioner (Room, on, heating).

Although rules concerning anomalies could be found, they are not always easy to express. The scalability and reusability requirements would not be met. Moreover, when taking into account an elderly person suffering from cognitive disorder, defining strict or complex rules may become extremely challenging. For these reasons, the second approach seems to be the most effective in our scenario. A probabilistic model is based on conditional probabilities that an event may occur depending on evidence or other variables. Expert knowledge could be elicited in conditional probability tables of the events (nodes) instead of rules. Furthermore, probabilistic models such as Bayesian Networks could be trained and thus tailored to different scenarios. This approach allows us to define a totally accessible and easy to understand system.

3.2.4.4 Bayesian Networks

A Bayesian network24 is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). Bayesian networks are ideal for taking an event that occurred and predicting the likelihood that any one of several possible known causes was the contributing factor. Efficient algorithms can perform inference and learning in Bayesian networks. For this reason, we have decided to employ the HUGIN Expert25 Bayesian Network platform that can perform both.

Judea Pearl has thoroughly explored probabilistic models and thus Bayesian Networks, and has shown that the existing processes for creating inferences are three

24 https://en.wikipedia.org/wiki/Bayesian_network
25 https://www.hugin.com

(Pearl, Glymour, & Jewell, Causal Inference in Statistics, 2016) (Pearl & Mackenzie, The Book of Why: The New Science of Cause and Effect, 2019):

1. *Deduction*
2. *Induction*
3. *Abduction*

Bayesian Networks can perform inferences through either a deductive or abductive process of reasoning. Meanwhile, the inductive process is the one used to find rules that shape the network. The differences between these two types of reasoning are explained below:

- *Deduction*, Deductive reasoning goes in the same direction as that of the conditionals, and links premises with conclusions. If all premises are true, the terms are clear, and the rules of deductive logic are followed, then the conclusion reached is necessarily true, and there is no *uncertainty*. Deductive reasoning allows deriving B from A only where B is a formal logical consequence of A. Deduction derives the consequences of the assumed. Given the truth of the assumptions, a deduction guarantees the truth of the conclusion. For example, given that the oven is on (A1), and the user has left home (A2), it follows that the user has forgotten the oven on (B). An example of a deductive Bayesian Network is shown in *(Fig. 31).* The *oven* is *On* for 10% of the time and the user leaves home for 30% of the time. It follows that the conditional probability that the user forgets the oven on after leaving his or her home is at 3%. Updating the evidence (100% oven On and 100% user has left home), the likelihood of the consequence will rise up to 100%.

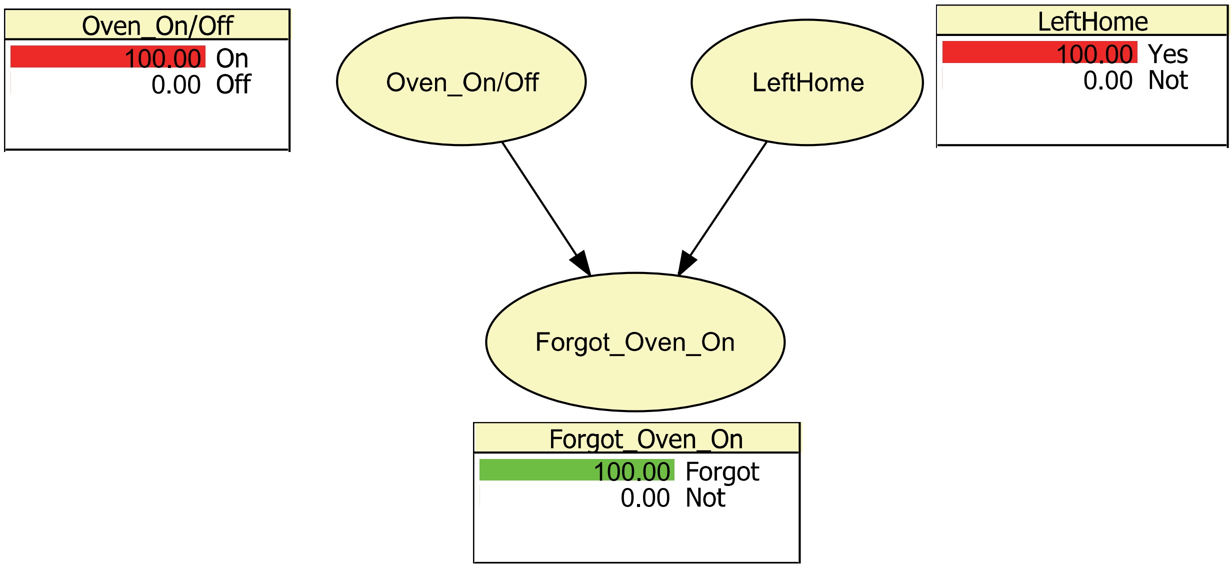


Figure 31, *Deductive reasoning in a Bayesian Network. Facts enable consequences to be inferred.*

- *Abduction*, Abductive reasoning allows inferring A as an explanation of B As a result of this inference, abduction allows the precondition A to be abducted from the consequence B. This approach is crucial when some variables or causes are unknown, and it enables diagnostics too. If we observe a consequence, we might imagine some causes in an abductive process. Abduction is formally equivalent to the logical fallacy of affirming the consequent because of multiple possible explanations for B. An example of abductive Bayesian Network is shown in *(Fig. 32)*.

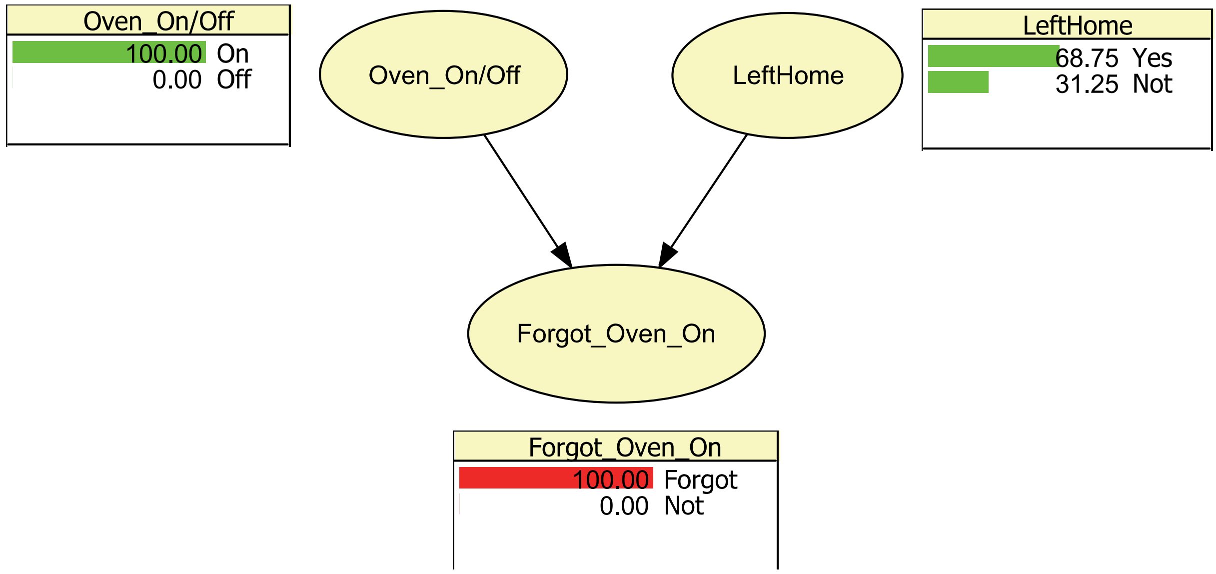


Figure 32, *Abductive reasoning in a Bayesian Network. A consequence is observed (the user has forgotten the oven on) and the likelihood of one of the possible causes (LeftHome) has risen to 68.75%.*

In our case, Bayesian Networks should be able to infer different types of anomalies:

- Anomalies that define wasteful situations (e.g., window open + heating on)
- Anomalies that derive from nonsense behaviours (e.g., skipping meals)
- Anomalies that define dangerous situations (e.g., something on the ground)
- Emergencies (e.g., falls)

Each of these possible circumstances have a different level of risk. It is indeed evident that the risk grows from low (wasteful situations and nonsense behaviours involve few risks) to high levels (e.g., emergency situations are determined by dangerous events such as falls, nausea, and so on).

To develop a BN for those purposes, we have firstly defined three different levels *(Tab. V)*. Each level represents a step forward in the development of knowledge. The first level consists of low-level knowledge data received from the readings of the sensors deployed in the smart home, the actions and activities inferred from the low-level knowledge model, and the direct interaction with the user through the dialog system. This level formally represents the *Evidence*. The third level includes the anomalies that could be found in the scenario where the system is deployed. It is the higher level of knowledge. The second level lies in the middle. It includes the possible consequences of the evidence (i.e., behaviours, feelings, intentions and general events that may happen in the environment). A system built upon these three levels of depth can perform either deductive or abductive reasoning. Deduction infers the probability that some anomalies may occur in real-time through a *top-down* process. Abduction allows a posteriori diagnostic that is useful to find out unobserved variables.

BNs levels		**Types of nodes**	**Level of knowledge**
Level 1	Evidence	*Sensor readings* *Actions inferred from low-level knowledge model* *Dialog System interaction*	Low
Level 2	Situations	*Behaviours* *Feelings* *Intentions* *Events*	High
Level 3	Awareness	*Anomaly Detection*	

Table V, The *underlying structure of the proposed Bayesian Network.*

Then, four modules have been formalized, always based on the structure shown above:

- *Nonsense and dangerous scenarios detector*
- *Behavioural changes detector*
- *Environmental comfort module*
- *Emergency module*

Since the same nodes may be found in more than one module (e.g., the node "*Feeling Cold"* may be located in both the Dangerous and Nonsense scenarios detector and the Environmental comfort model), we have developed a specific type of BN, namely *Object-Oriented Bayesian Network (OOBN)*. We have intentionally avoided dynamic networks due to the variability of possible scenarios in an AAL environment. OOBNs can deal with complex systems due to their structure and rely on scalable sub-modules that can be reused. The reusability of the modules will also allow the creation of module libraries in later phases.

3.2.4.5 Automation HUB

Since our Digital Twin requires interaction with the real environment and the user in order to deliver AAL services, we decided to perform such interaction through a *flow-based programming tool*. In computer programming, flow-based programming (FBP) is a programming paradigm that outlines applications as networks of *black box* processes, which exchange data across predefined connections by message passing, where the connections are specified externally to the processes. These black box processes can be reconnected endlessly to form different applications without having to be changed internally. FBP is thus naturally component-oriented (Wikipedia, Flow-based programming, 2021).

Current developments try to integrate dataflow programming languages with the visual programming approach to either have direct access to the program state, resulting in online debugging, or automatic program generation and documentation. *Visual Programming Language* (VPL) is any programming language that lets users create programs by manipulating program elements (black boxes) graphically rather than by specifying them textually. A VPL allows programming with visual expressions, spatial arrangements of text and graphic symbols, used either as elements of syntax or secondary notation. For example, many VPLs (known as dataflow or diagrammatic programming) are based on the idea of "boxes and arrows", where boxes or other screen objects are treated as entities, connected by arrows, lines or arcs which represent relations (Wikipedia, Visual programming language, 2021).

VPLs may be further classified, according to the type and extent of visual expression used, into icon-based languages, form-based languages, and diagram languages. Visual programming environments provide graphical or iconic elements which can be manipulated by users in an interactive way according to some specific spatial grammar for program construction.

The general goal of VPLs is to make programming more accessible to novices and to support programmers at three different levels:

- *Syntax*, VPLs use icons/blocks, forms and diagrams trying to reduce or even to completely eliminate the potential of syntactic errors helping with the arrangement of programming primitives to create well-formed programs.
- *Semantics*, VPLs may provide some mechanisms to disclose the meaning of programming primitives. This could include help functions providing documentation functions built-in to programming languages.
- *Pragmatics*, VPLs support the study of what programs mean in particular situations. This level of support allows users to put artifacts created with a VPL into a certain state in order to explore how the program will react to that state.

Node-RED[26] is a FBP tool for wiring together hardware devices, APIs and online services through a VPL. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click. It is a model that lends itself very well to a visual representation and makes it more accessible to a wider range of users. If someone can break down a problem into discrete steps, they can look at a flow and get a sense of what it is doing; without having to understand the individual lines of code within each node. Node-RED consists of a *Node.js* based runtime that you point a web browser at to access the flow editor. Within the browser you create your application by dragging nodes from your palette into a workspace and start to wire them together. With a single click, the application is deployed back to the runtime where it is run. The palette of nodes can be easily extended by installing new nodes created by the community (Node-RED, s.d.).

26 https://nodered.org

3.2.5 Service Layer

The Service Layer holds the purposes of the Digital Twin. Since the model aims to find out anomalies and support the user in his or her ADLs, the proposed services are related to those aims. The services required are actuated by the Automation HUB. Specifically, the home automation platform links the results yielded from the BNs to the dialog system, the devices and the actuators integrated in the smart environment. Three main services have been proposed in our model:

- *Emergency call*, When the emergency OOBN module infers dangerous situations caused by actions such as falling down and so on, the emergency call service must alert first aid and primary carers.
- *Support without interaction (actuators)*, In some scenarios the user may not respond to interactions. In these cases the actuators are triggered to avoid dangerous situations (e.g., the user has left home and forgot the cooktop turned on, the system trigger the actuator that will turn off the cooktop to avoid fires).
- *Support though interaction*, The dialog system interacts with the user to remind things, communicate system's actions, alert of an anomaly, and so forth, depending on the scenario inferred from the BN.

CHAPTER IV

RESEARCH DEVELOPMENT

This chapter includes the developments of the research. As the system presented in Chapter 3 consists of several components, the individual parts of the system were developed and tested individually. It has not been possible to fully test the entire pipeline for technical and time related aspects, as this is a master's degree dissertation. Nevertheless, the validation of the single parts will ensure the correct functioning of the whole system since gaps between the different elements can be easily addressed and overcome with rather basic technical formalizations. Thus, the following paragraphs present the work conducted to test and validate different aspects of our system:

- *Real-time 3D virtual representation (4.1).*
- *Low-level knowledge models: Neural Networks for Activity Recognition (4.2).*
- *High-level knowledge model: Object-Oriented Bayesian Networks, (4.3).*
- *Home automation and vocal interaction with the user (4.4).*

4.1 Real-time 3D virtual representation

Digital Twins mirror real environments creating virtual instances of the asset. The virtual representation of scenarios is based on static and dynamic real-time data. For our research we have embedded the user's avatar into a kitchen environment. The results of both integration and stabilization of data regarding the building and the user into the game engine are shown in *(Fig. 33)*.

Figure 33, *Frames extracted from the game engine runtime showing the user's avatar moving in real-time in an arbitrary kitchen environment.*

4.1.1. Data integration

The integration among the BIM model, the Skeleton Tracking model and the game engine is organised in two different phases. The first phase concerns the integration of BIM objects that provide information about the building (in our case study a single unit) and its components. In the second phase, the user's avatar is integrated into the virtual scenario and therefore recognized within the 3D model of the apartment with its real-time exact pose and orientation.

First, the BIM model has been created in Revit. Since objects have been downloaded from an open BIM repository, they already include all the IFC27 information. Usually, the Revit model is converted to an IFC model to be imported in the game engine while maintaining all the information about its components. Nevertheless, we have intentionally avoided this process as it is time-consuming and not convenient for this research. It is worth noting that we are aware that a future implementation will require the use of the IFC file format to ensure interoperability, but it is not the main purpose of this work. Hence, the BIM model has been exported in the FBX format instead of IFC to validate our model. To preserve objects materials after the conversion, the Twinmotion Revit plugin28 is exploited. Once exported the FBX file, the model is imported in the game engine *(Fig. 34)*.

27 https://technical.buildingsmart.org/standards/ifc/ifc-formats
28 https://www.unrealengine.com/en-US/twinmotion/plugins

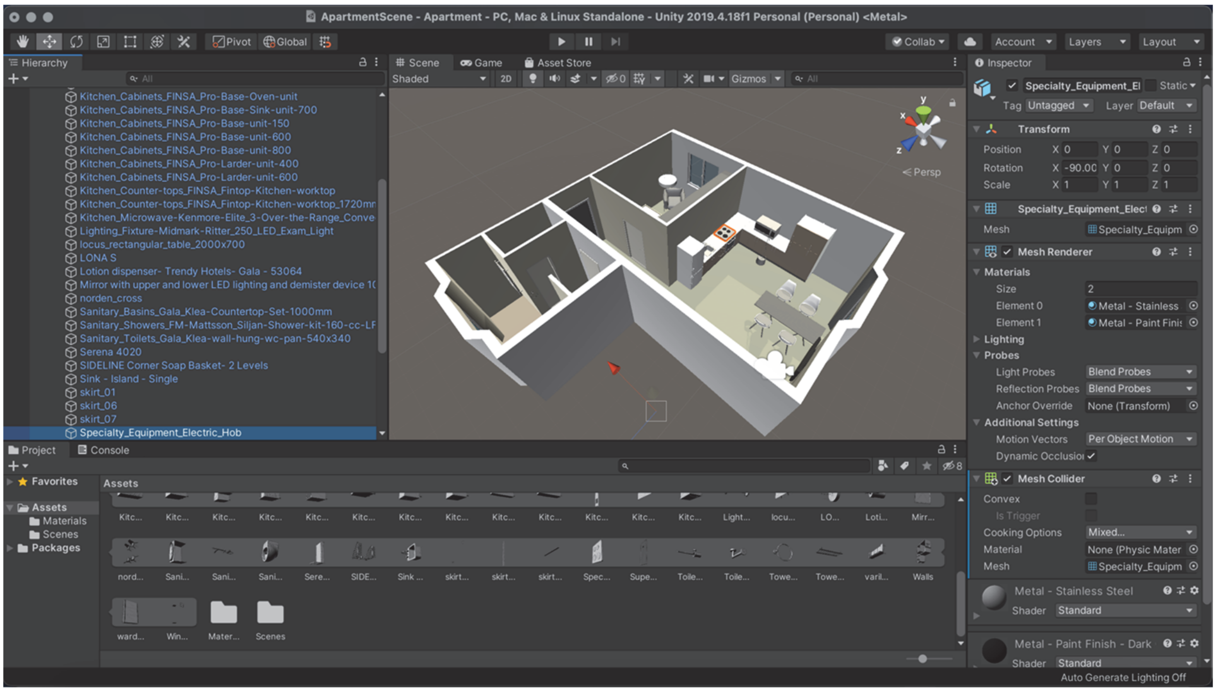


Figure 34, *Asset imported into the game engine. The import of the FBX format recognizes all the BIM objects (to the left) and imports them as prefabs (bottom), which are replicable and modifiable directly into this platform. Furthermore, prefabs preserve their own material. Mesh collider has been applied to all the tangible components to avoid inconsistencies when the user's avatar will be added into the scene.*

Then, the Skeleton Tracking algorithm is integrated into the game engine to achieve the real-time 3D virtual representation of the user in the environment. The Nuitrack AI algorithm is used in this phase. It exploits 3D data coming from the integrated visual sensor (Intel RealSense L515) as input to detect the user's skeleton. The distance between the user and the camera should be unobstructed and not exceed 4/5 metres to have consistent results. The camera has been positioned at a height of 1 metre and levelled horizontally. The result of the Skeleton Tracking model is shown in *(Fig. 35)*.

Figure 35, *Testing the Skeleton Tracking algorithm into the game engine. The user's movements are mirrored in real-time by the avatar placed into the virtual scenario.*

4.1.2 Skeleton stabilization

The virtual avatar of our user was reporting errors in its positions, scale and movements. These issues are caused by the low quality of raw data. Although we use a high-quality LiDAR camera (i.e., Intel RealSense L515), and the Skeleton Tracking algorithm (Nuitrack AI in our case) is a powerful AI tool, it can face difficulties in tracking the user's movement in situations where the view is not clear, the view of the skeleton is not complete, the distance exceeds a certain threshold. A first level of filtering *(Algorithm 1)* is applied to the confidence of the data derived from skeleton tracking. The confidence is automatically produced by Nuitrack AI and is a percentage value. Some joints may have low confidence values, meaning that they are not very reliable. If this data were not cleaned, it would lead to distortions of the avatar and bias caused by non-natural joint positions. To overcome this problem, a confidence threshold has been introduced with a value of 0.1. The avatar will therefore not consider data from joints whose confidence is below 10%.

Algorithm 1 Filtering non-confident data

Input: Raw data from skeleton tracking

Output: Confident data from skeleton tracking

Define a threshold:

float thr = 0.1f;

Scan all avatar joints and move avatar based on skeleton detected by Nuitrack:

foreach *(var joint in jointsMapping)*

  Get corresponding joint from the Nuitrack:

  nuitrack.Joint jointNuitrack = skeleton.GetJoint(joint.Key);

  Check if confidence is greater than threshold:

  if *(joinNuitrack.Confidence ≥ thr)* ***then***

   Get joint of the avatar corresponding to selected nuitrack joint:

   ModelJoint avatarJoint = joint.Value;

   Apply stabilization

  end if

end foreach

A second level of filtering *(Algorithm 2)* is applied to solve the stabilization issue. We have developed an algorithm which filters the incoming data, that comes from the 3D skeleton tracking model, to have more natural and smooth movements, accurate position in the space, and cleaner output data that will consequently feed the Neural Networks. The algorithm is based on the following equation:

$$X_{(t+1)} = (1 - a) \cdot X_{(t)} + a \cdot X_{(t+1)}^{raw} \qquad (13)$$

where:

- $X_{(t+1)}$ is the processed data at the time $(t + 1)$.
- $X_{(t)}$ it the processed data at the time t.
- a is a corrective factor with a value that ranges between 0 and 1 (very close to 0). Consequently, $(1 - a)$ is close to 1, and $a \cdot X_{(t+1)}^{raw}$ is close to 0.
- $X_{(t+1)}^{raw}$ is the raw value of the data at the time $(t + 1)$.

This filter has been applied to improve three features of the avatar:

1. *Height*, the height of the avatar is calculated by summing the lengths of the bones (found by difference from the coordinates of the joints). The summed bones are the ones from the foot to the hip and from the pelvis to the head. A further correction factor has been added. It multiplies the neck-head distance by 2.4 because the head joint is not located at the top of the head but at the level of the nose. This makes the height of the avatar very close to that of the user.
2. *Position*, the position of the whole skeleton relies on the position of the root joint (torso). A more accurate precision of such a joint means better positioning of the avatar into the environment.
3. *Orientation of its bones*, each bone has its own orientation that is based on relative coordinates (position and rotation).

While height and position of the avatar both concern the *x, y, z* values, the orientation of the bones is also related to their rotations. For this reason, although the structure of the filter *(Eq. 13)* remains the same, the bones orientation algorithm is based on the function *quaternion.slerp* in order to define an average rotation between the times (t) and $(t+1)$. The quaternion formulation is equivalent to that expressed above and allows consistency in the rotation to be maintained. It is also a more rigorous and less complex approach.

Algorithm 2 Stabilization of the skeleton

Input: Confident data from skeleton tracking

Output: Stabilized avatar

Find Skeleton

Define weights:

float a = 0.02f;

float t = 0.2f;

if *(skeleton.GetJoint(nuitrack.JointType).Confidence ≥ thr)* ***then***

Sum of the differences between the *z-values* of the joints to find *height*

Apply filter to *height*:

*height = (1-a) * height + a * newHeight;*

Scale avatar to match human height:

targetGameObject.transform.localScale = new Vector3(height/1.80f, height/1.80f, height/1.80f);

Calculate the model position:

*Vector3 rootPos = Quaternion.Euler(0f, 180f, 0f) * (0.001f * skeleton.GetJoint(rootJoint).ToVector3());*

Apply filter to *position*:

*transform.position = (1-t) * transform.position + t * rootPos;*

Calculate the model bone rotation:

*Quaternion jointOrient = Quaternion.Inverse(CalibrationInfo.SensorOrientation) * (jointNuitrack.ToQuaternionMirrored()) * avatarJoint.baseRotOffset;*

Apply rotation to bone based on the detected skeleton:

avatarJoint.bone.rotation = Quaternion.Slerp(avatarJoint.bone.rotation, jointOrient, t);

end if

The results are shown in *(Fig. 36)*. Whereas the initial avatar seems to develop inconsistencies in some movements, the processed one maintains its stability. It is worth noting that weights and thresholds in the developed algorithm could be modified as desired to achieve the most effective result.

Figure 36, *Comparison between non-processed (left) and processed (right) avatars.*

4.2. Neural Networks for Activity Recognition

In our Digital Twin, Neural Networks have been chosen to extract information regarding the type of action or activity the user is currently pursuing in the AAL environment. The reason lies in the fact that these types of models are appropriate for this role. Indeed, a black box analytical model such as neural networks is capable of extracting useful information, such as activities, from raw data, such as skeleton coordinates. A model of this nature can be defined as low-level because it does not elicit information through contextualization, but only through a series of analytical computations (which take place in artificial neurons) whose intermediate results are definitely meaningless. Neural networks are therefore capable of extracting from the source data a single type of information, in our case the user's activities, depending on the type of network and the dataset on which the network itself is trained.

The NN deployed in our system an open model, namely *MS-G3D*, which is pre-trained on the NTU RGB+D 60 dataset. This dataset includes sixty human actions/activities. It consists of 40 daily actions, 9 actions regarding medical conditions, and 11 mutual actions, all shown in *(Tab. VI)*. Since the system aims at supporting a single user, the daily and medical condition actions only have been taken into account at the current stage. The dataset has different sample modalities:

- *3D skeleton (body joints)*
- *Masked depth maps*
- *Full depth maps*
- *RGB videos*
- *IR data*

Since we extract 3D data from our own avatar, the pre-trained model chosen29 is the one trained on the NTU RGB+D 60 Skeleton dataset. This allows us to have a very lightweight model as well. While the other sample modalities of the dataset have a size ranging from 83 Gb to 886 Gb, the 3D skeleton mode has a size of only 5.8 Gb.

29 MS-G3D pre-trained models available at https://github.com/kenziyuliu/MS-G3D

NTU RGB-D 60 dataset								
N	**Code**	**Actions/Activities**	**N**	**Code**	**Actions/Activities**	**N**	**Code**	**Mutual actions**
1	A1	drink water	**26**	A26	hopping	**50**	A50	punch/slap
2	A2	eat meal	**27**	A27	jump up	**51**	A51	kicking
3	A3	brush teeth	**28**	A28	phone call	**52**	A52	pushing
4	A4	brush hair	**29**	A29	play with phone/tablet	**53**	A53	pat on back
5	A5	drop	**30**	A30	type on a keyboard	**54**	A54	point finger
6	A6	pick up	**31**	A31	point to something	**55**	A55	hugging
7	A7	throw	**32**	A32	taking a selfie	**56**	A56	giving object
8	A8	sit down	**33**	A33	check time (from watch)	**57**	A57	touch pocket
9	A9	stand up	**34**	A34	rub two hands	**58**	A58	shaking hands
10	A10	clapping	**35**	A35	nod head/bow	**59**	A59	walking towards
11	A11	reading	**36**	A36	shake head	**60**	A60	walking apart
12	A12	writing	**37**	A37	wipe face			
13	A13	tear up paper	**38**	A38	salute			
14	A14	put on jacket	**39**	A39	put palms together			
15	A15	take off jacket	**40**	A40	cross hands in front			
16	A16	put on a shoe	**41**	A41	sneeze/cough			
17	A17	take off a shoe	**42**	A42	staggering			
18	A18	put on glasses	**43**	A43	falling down			
19	A19	take off glasses	**44**	A44	headache			
20	A20	put on a hat/cap	**45**	A45	chest pain			
21	A21	take off a hat/cap	**46**	A46	back pain			
22	A22	cheer up	**47**	A47	neck pain			
23	A23	hand waving	**48**	A48	nausea/vomiting			
24	A24	kicking something	**49**	A49	fan self			
25	A25	reach into pocket						

Table VI, *List of actions/activities included in the NTU RGB+D 60 dataset. "Mutual actions" refer to two-people interactions. The red actions trigger emergency call, the orange ones could trigger a report that could be checked by general practitioners.*

The following chapters explain the process followed to implement the MS-G3D model in our system: first, we need to prepare the input data regarding the joints coordinates; then, the system is implemented. It is worth noting that training and testing on the dataset are omitted since the model has already been pre-trained and tested on the NTU dataset by its developers in (Liu, Zhang, Chen, Wang, & Ouyang, 2020).

4.2.1. Mapping Avatar Joints

The avatar into the game engine is moved through the 19 filtered joints information. Nevertheless, the avatar chosen has a greater number of joints (25). It means that 6 of the 25 joints do not receive any kind of information. Although the input information is therefore incomplete, this does not cause problems in both the representation of the avatar and the export of the coordinates. In fact, the joints that do not receive data, besides being not relevant joints (e.g., fingertips, etc.), will move together with the rest of the body in a natural way. Thus, their coordinates can be subsequently exported maintaining a certain level of consistency. To do so, we have mapped the joints of the avatar *(Algorithm 3)*. The 25 mapped joints are the same as those in the NTU RGB+D dataset. Hence, the NN pre-trained on that dataset are ready to be fed by real-time data from our own processed skeleton. It is worth noting that since the order of the joints is relevant for the recognition task, they have been organized in the same order as in the NTU RGB+D dataset.

Algorithm 3 Mapping Avatar Joints

Input: Skeleton Joints

Output: New Mapped Avatar Joints

Define the new List of Joints:

```
readonly Dictionary<int, string> exportList = new Dictionary<int, string>()
  {
    {1 , "Root_M"},
    {2 , "Spine1_M"},
    {3 , "Neck_M"},
    {4 , "Head_M"},
    {5 , "Shoulder_L"},
    {6 , "Elbow_L"},
    {7 , "Wrist_L"},
    {8 , "MiddleFinger1_L"},
    {9 , "Shoulder_R"},
    {10 , "Elbow_R"},
    {11 , "Wrist_R"},
    {12 , "MiddleFinger1_R"},
    {13 , "Hip_L"},
    {14 , "Knee_L"},
    {15 , "Ankle_L"},
    {16 , "Toes_L"},
    {17 , "Hip_R"},
    {18 , "Knee_R"},
    {19 , "Ankle_R"},
    {20 , "Toes_R"},
    {21 , "Scapula_M"}, // It is computed as mean between Scapula_L & Scapula_R
    {22 , "MiddleFinger4_L"},
    {23 , "ThumbFinger4_L"},
    {24 , "MiddleFinger4_R"},
    {25 , "ThumbFinger4_R"},
  };
```

4.2.2 Exporting Avatar Joints information

Since the NN model is pre-trained on the NTU RGB+D dataset, the avatar frame sequences should be transformed into real-time data concerning its joint coordinates in order to be correctly read by the NN. To this end, an algorithm has been developed to update a text file in real-time containing data on the coordinates of the avatar joints for each frame *(Algorithm 4)*.

Algorithm 4 Exporting real-time text file with Avatar Joints coordinates

---

Input: Coordinates of the 25 mapped joints of the avatar

Output: Text file updated in real-time with coordinate matrixes of the avatar joints

```
Access to the new joint list:
log skeleton data in text file
  StreamWriter writer = new StreamWriter(logFile, true);
 Define number of bodies:
  writer.WriteLine("1");
Define number of joints:
  writer.WriteLine("25");
Export matrix:
  foreach (var item in exportList)
  Vector3 pos = new Vector3(float.NaN, float.NaN, float.NaN);
    try
        GameObject go = GameObject.Find(item.Value).gameObject;
        pos = go.transform.position;
    catch (Exception)
        if (item.Value.Right(1) == "M")
            pos = GameObject.Find(item.Value.Left(item.Value.Length - 1) +
            "L").gameObject.transform.position;
            pos += GameObject.Find(item.Value.Left(item.Value.Length - 1) +
            "R").gameObject.transform.position;
            pos /= 2;
        end if
        if (pos.x != float.NaN)
            writer.WriteLine(pos.x.ToString(nfi) + " " + pos.y.ToString(nfi) + " " +
            pos.z.ToString(nfi) + " 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 2");
```

```
        else
            writer.WriteLine("0 0 0 NaN NaN NaN NaN 0 1 0 0 0");
        end if
    end foreach
writer.Close();
```

The structure of each NTU RGB+D skeletal frame is shown in *(Tab. VII a)*. As the majority of the information contained in that table are related to the Kinect v2 tool, we choose to discard those additional data without compromising the Activity Recognition task. Consequently, our coordinate matrix has less data as shown in *(Tab. VII b). As a matter of fact, there are only x, y and z coordinates of each joint (columns highlighted with bold edges), whereas all the Kinect v2 related data have been discarded (set to zero).* Frame by frame, the coordinate matrix is added to its preceding ones. In this way, the Neural Network system can be fed with a sequence of frames (300 frames) that each activity recognition computation requires.

		ID											
		103											
		N° of bodies											
		1											
		bodyID	**clipedEdges**	**handLeftConfidence**	**handLeftState**	**handRightConfidence**	**handRightState**	**isResticted**	**leanX**	**leanY**	**trackingState**		
		72057594037931101	0	1	1	1	1	0	0,02764709	0,05745083	2		
		jointCount											
		25											
		joints											
connecting_joint	**#**	**x**	**y**	**z**	**depthX**	**depthY**	**colorX**	**colorY**	**orientationW**	**orientationX**	**orientationY**	**orientationZ**	**trackingState**
2	**1**	0,2181153	0,1725972	3,785547	277,419	191,8218	1036,233	519,1677	-0,2059419	0,05349901	0,9692109	-0,1239193	2
1	**2**	0,2323292	0,4326636	3,714767	279,2439	165,8569	1041,918	444,3235	-0,2272637	0,05621852	0,964434	-0,1227094	2
21	**3**	0,2457799	0,6877249	3,633897	281,1529	139,0885	1047,837	367,3966	-0,2486043	0,0660736	0,9535829	-0,1565561	2
3	**4**	0,2128507	0,8079225	3,581995	278,1617	125,6969	1039,476	328,9554	0	0	0	0	2
21	**5**	0,1109304	0,6111551	3,716962	267,2831	148,2511	1007,479	393,5405	0,2179059	0,7353331	-0,6371263	-0,07663021	2
5	**6**	0,100875	0,4286715	3,742593	266,2123	166,5774	1004,187	446,2603	0,03366299	0,7429995	-0,0668001	-0,6650987	1
6	**7**	0,1186992	0,3428923	3,556542	268,5588	173,2219	1011,711	465,4302	-0,5255184	0,8340511	0,1235747	-0,1136605	1
7	**8**	0,1165352	0,3173672	3,550741	268,3543	175,7982	1011,131	472,8558	-0,1148792	0,9877279	-0,0299019	-0,1014999	2
21	**9**	0,3449145	0,5740387	3,588655	291,5822	149,8556	1078,05	398,4594	-0,1106443	0,771432	0,5446612	-0,3098302	2
9	**10**	0,4211396	0,3847547	3,627778	298,8716	169,6459	1098,838	455,4754	0,06063642	0,961247	0,2020273	0,1775176	2
10	**11**	0,1918999	0,3189077	3,540756	276,1781	175,5411	1033,77	472,192	0,4641625	0,2649592	-0,3935434	0,7479796	2
11	**12**	0,1298675	0,317039	3,542577	269,7591	175,7559	1015,225	472,7473	0,6194489	0,3464894	-0,3181734	0,6284853	2
1	**13**	0,162766	0,1745395	3,778471	272,0995	191,6048	1020,896	518,4943	-0,05136763	-0,6810405	0,7174643	-0,1370778	2
13	**14**	0,1975115	-0,1360554	3,904381	274,8451	221,2371	1028,12	604,0775	-0,1458015	-0,5206363	0,1335629	0,8305664	2
14	**15**	0,2442706	-0,4201995	4,054147	278,3998	246,4292	1037,602	676,7612	-0,04763737	0,1068644	0,244998	0,9624379	2
15	**16**	0,2040377	-0,476396	4,094294	274,5897	251,0867	1026,418	690,1627	0	0	0	0	2
1	**17**	0,2699234	0,1678491	3,730005	282,8167	192,0363	1052,036	519,8357	-0,2298583	0,6472055	0,644957	-0,3351428	2
17	**18**	0,3105748	-0,1462299	3,818383	286,1002	222,5039	1060,931	607,8004	0,08184162	0,837252	0,1252462	0,5259509	2
18	**19**	0,3428889	-0,433214	3,943258	288,1872	248,7218	1066,199	683,3955	0,2016301	0,9774966	0,05738356	0,02351441	2
19	**20**	0,2951482	-0,5017325	3,968896	283,5814	254,7903	1052,752	700,8542	0	0	0	0	2
2	**21**	0,2425592	0,6247278	3,656055	280,6673	145,8599	1046,329	386,8308	-0,2489336	0,06207102	0,9568546	-0,1364125	2
8	**22**	0,1200176	0,2955468	3,532409	268,7763	177,8907	1012,419	478,8939	0	0	0	0	2
8	**23**	0,0989792	0,3459614	3,527539	266,6139	172,6136	1006,227	463,6581	0	0	0	0	2
12	**24**	0,08710064	0,2988889	3,54619	265,3322	177,6662	1002,409	478,2144	0	0	0	0	2
12	**25**	0,1268453	0,2813963	3,549525	269,4193	179,4981	1014,19	483,5357	0	0	0	0	2

Table VII a, *Example of the NTU RGB+D Skeletal frame file structure.*

503											
1											
11111111111111100	0	1	1	1	1	0	0.0	0.0	2		
25											
0.01791066	1,0075170	-2,052639	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.0531436	1,1693360	-2,009135	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.01530945	1,5147950	-2,134020	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.04892847	1,5915100	-2,080812	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
-0.1328553	1,3609720	-2,029415	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
-0.228662	1,1077700	-2,065163	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
-0.3867974	0,9150124	-2,017069	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
-0.4753607	0,8586560	-1,996326	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.1327867	1,5169040	-2,237203	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.1743356	1,2631680	-2,329175	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.3432376	1,0815860	-2,383585	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.4306612	1,0311180	-2,419080	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
-0.07658753	0,9965007	-2,015729	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
-0.2455689	0,5838541	-2,073565	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
-0.4354308	0,2600926	-2,209658	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
-0.4036545	0,1318449	-2,099440	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.100265	0,9826782	-2,107543	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.122242	0,5338213	-2,122427	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.0954127	0,1492995	-2,226415	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.1626806	0,0565983	-2,098015	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.01358407	1,4356590	-2,125429	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
-0.512163	0,7766080	-1,952750	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
-0.4541804	0,8676810	-1,902087	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.4915819	0,9530635	-2,405612	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
0.4684294	1,0427430	-2,330482	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2

Table VII b, *Example of our skeletal frame file structure. It represents all the C* dimensional feature *vector of each node $(v_1, \ldots, v_{25})$, namely $\boldsymbol{X}_{t,n}$ at frame time t over a total of T frames, with $\mathbf{X}_{t,n} \in \mathbb{R}^{N \times C}$. Each line of the table represents a different avatar joint, the first three edged columns represent x, y, and z coordinates of each joint at time t.*

4.2.3 MS-G3D system implementation

As explained in *(Chapter 3.2.4.2)*, the MS-G3D system should be fed with a skeleton matrix that includes joints coordinates. This feature vector is expressed as $\mathbf{X} \in \mathbb{R}^{T \times N \times C}$. $\mathbf{X}$ includes the number of joints (N) and their coordinates (C) throughout the time (T). An example of a feature vector is shown in *(Table VII b)*. The model requires a feature vector that provides information for a time of 10 seconds ($T = 300$ if 30 fps).

However, the NTU RGB+D dataset, on which the model has been trained, fed the network with frame sequences that last 2 seconds. These sequences are then multiplied by 5 to obtain a duration of 10 seconds. This is because 2 seconds is a rather standard duration of an action. Longer durations would create ambiguity and inconsistency in the action to be recognised, which in a longer time could be more than one. Shorter lengths of sequence would also not allow the system to perform its function correctly, because the action may not be still completed if the length of the sequence is too short. Furthermore, to ensure reliable performances, we feed the network with frame sequences 2 seconds long and repeated 5 times. Those are extracted each second to prevent us from not capturing actions that might take place in the middle of each 2-second sequence. The algorithm below has been developed to show the 3D skeleton by extrapolating the avatar joints coordinates from the skeleton file containing the feature vector *(Algorithm 5)*. In *(Fig. 37)* to the left are shown some frames of a skeletal sequence obtained by running the script below.

Algorithm 5 Show 3D Skeleton

Input: Skeleton Joints coordinates
Output: 3D Skeleton visualization

```
Read text file with joints coordinates:
  bodyinfo = read_skeleton_file('joints_log_1.skeleton');
Each joint is connected to some other joint:
  connecting_joint = ...
     [2 1 21 3 21 5 6 7 21 9 10 11 1 13 14 15 1 17 18 19 2 8 8 12 12];
For every frame:
  for f=1:numel(bodyinfo)
  For all the detected skeletons in the current frame:
       for b=1:numel(bodyinfo(f).bodies)
       For all the 25 joints within each skeleton:
          for j=1:25
          Connect joints:
              try
              k = connecting_joint(j);
              joint = bodyinfo(f).bodies(b).joints(j);
              x1(j) = joint.x;
              y1(j) = joint.z;
              z1(j) = joint.y;
              joint2 = bodyinfo(f).bodies(b).joints(k);
```

```
                x2(j) = joint2.x;
                y2(j) = joint2.z;
                z2(j) = joint2.y;
                catch err1
                disp(err1);
            end
        end
    end
end
```

At that point, the MS-G3D model executes its computation on each feature vector. The *main.py* script of the NN model *(Algorithm 6)* is shown in *(Appendix A)*. Its output are the scores of each activity, i.e., the percentage values of each from the least probable to the most probable. The whole implementation process is shown in *(Fig. 37)*.

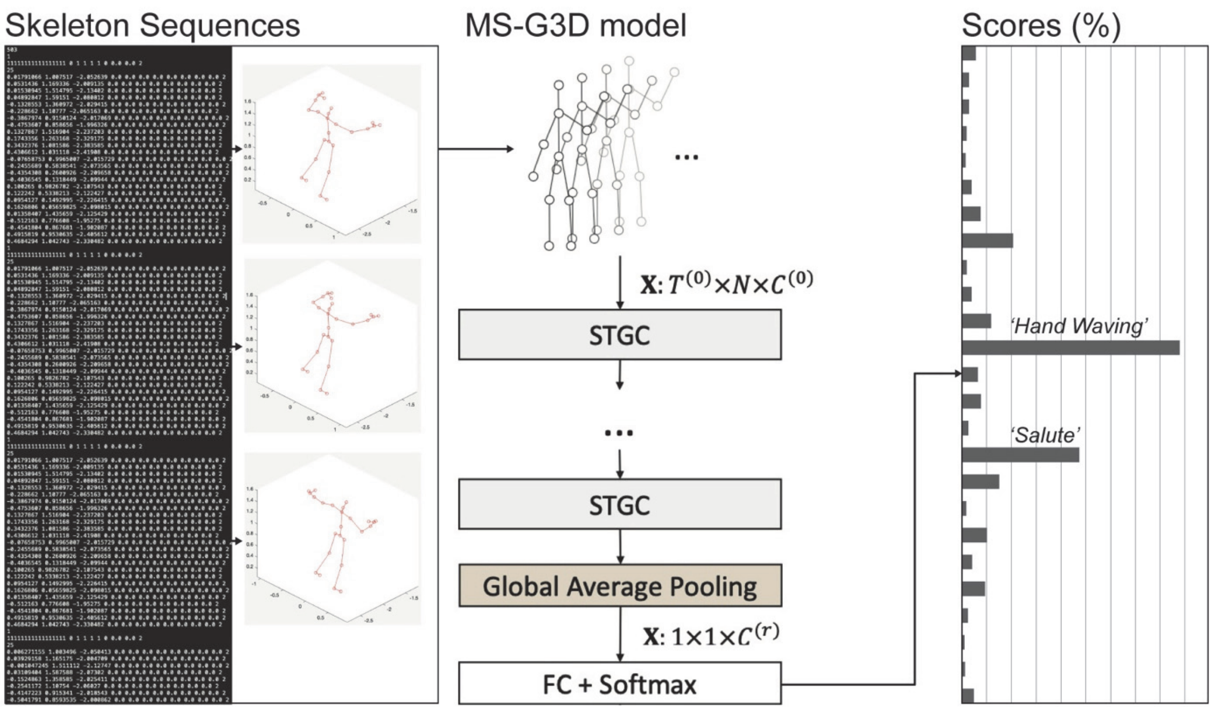


Figure 37, *Activity Recognition process using the MS-G3D Neural Network model. From skeleton sequences to activity's scores.*

4.3. Object-Oriented Bayesian Networks

In the proposed system, the Object-Oriented Bayesian Network works as a High-level knowledge model. It is the Reasoner of the whole system, due to its features. A new semantic regarding probable situations, scenarios, and anomalies in AAL has been built to formalize the networks. This kind of anomaly recognition model can be defined as a high-level knowledge model since the reasoning is based on information regarding the context. Indeed, high-level reasoners are those that allow contextualization. In such cases, the results (decisions) are not carried out through some kind of analytical black box process (as the NN does), but associating scenarios to more basic semantics (actions, history, environment). This process is called contextualization and defines readable and accessible models. Bayesian Networks are an extremely valuable tool that enables contextualization.

4.3.1. Definition of recognizable scenarios

There are different mental illnesses, and they affect cognition in different ways. Furthermore, not every person is affected in the same way. Some people with schizophrenia have more cognitive problems than others. Some people with depression or bipolar disorder may have problems in one aspect of cognitive functioning but not another. It is important to understand that a mental illness affects each person fairly differently (Medalia & Revheim, 2002). Defining a generalized and scalable method to cope with each possible situation that may occur in an environment inhabited by a user that suffers from cognitive disorders is crucial, even though it is fairly challenging.

Firstly, this approach should take into account general situations that could lead to dangerous or nonsense scenarios. As explained in (scie.org, 2020), those scenarios are usually caused by:

- *Confusion*
- *Depression*
- *Loss of memory*
- *Emotional distress*

- *Difficulty paying attention*

A list of general dangerous and nonsense scenarios that may occur in an AAL environment is shown in *(Tab. VIII)*. The scenarios are symptoms of cognitive disorders and they have been separated in two main categories:

- *General scenarios* (Pirani, et al., 2020)
- *Cooking/Eating scenarios* (Gillespie, 2019)

Highlighted rows in the table below represent circumstances which our system is able to recognize. Thus, these scenarios have been called *Recognizable*, whereas others are marked as *non-Recognizable* due to the *Additional missing requirements* explained beside each of them. Nevertheless, the recognition of all other scenarios is related to merely technical factors that would require not too much effort. Such missing requirements are:

- *Training the Bayesian network on real data in a real AAL scenario*, given the inconsistency and variability of data regarding this kind of scenario, and not being able to extract data by testing our system in a real AAL environment, training with real data has not been viable.
- *Type of action,* the tested neural networks have been pre-trained on the NTU-RGB+D 60 dataset which supports 60 types of user action/activity. This limitation does not allow us to detect particular scenarios based on specific activities such as medicating.
- *Storage of activity time patterns, Dialog System interactions, and system interventions (actuations)*, to recognize some situations it would be necessary to store data. This involves developing time-pattern systems that report activations of activity labels, dialog system interactions, and actuators.
- *Object recognition and position*, detecting objects is a task performed by state-of-the-art NNs *(Chapter 2.4.1.1)* that have not been integrated in our system yet.
- *Deployment of Ontologies*, procedures, instructions, recipes, and directions could be extracted by ontologies. An example of recipe ontology is the one defined in (Salvador, et al., 2019).

However, the accessibility of our system allows further features to be integrated even in future stages, in order to recognize all the scenarios listed below.

	General Dangerous or Nonsense scenarios	Recognizable scenarios	Additional missing requirements
1	*Repetitions*	No	Storage of activity time patterns
2	*Slowing daily activities*	No	Training BN on real data
3	*Time disorientation*	Yes	_
4	*Difficulties arranging*	Yes	_
5	*Forgetting to take medications*	No	Type of Action
6	*Taking too many medications*	No	Type of Action + Storage of activity time patterns
7	*Cannot find items around the house*	No	Object recognition and position
8	*Repeating questions over and over*	No	Storage of Dialog System interactions
9	*Forgetting procedures*	No	Procedure Ontologies
10	*Cannot remember directions/instructions*	No	Procedure Ontologies
11	*Indifferent to the environment*	Yes	_
12	*Getting distracted in the middle of things*	No	Procedure Ontologies
13	*Frequently saying "I'm bored"*	No	Storage of Dialog System interactions
14	*Trying to do too many things simultaneously*	Yes	_
15	*Getting easily overwhelmed*	Yes	_
16	*Repeating mistakes with apparent learning from previous errors*	No	Storage of system interventions
17	*Finding trouble getting things started independently*	No	Training BN on real data + Storage of activity time patterns
18	*Does not like routines change*	No	Training BN on real data + Storage of activity time patterns
19	*Do not ask for help*	No	Training BN on real data + Storage of activity time patterns
20	*Does not finish what is started*	No	Procedure Ontologies
21	*Lazy to figure things out*	No	Procedure Ontologies
22	*Does not seek out alternatives*	No	Procedure Ontologies
23	*Immobilised*	No	Storage of temporal information

	Dangerous or Nonsense scenarios while cooking/eating	Recognizable scenarios	Additional missing requirements
24	*Struggling with a recipe*	No	Recipe Ontologies
25	*Getting mixed up while cooking*	No	Recipe Ontologies
26	*Forgetting steps*	No	Recipe Ontologies
27	*Mistaken measurements*	No	Recipe Ontologies
28	*Mishandling appliances*	Yes	_
29	*Spoiled foods*	No	Storage of temporal information
30	*Miscalculating cooking time*	No	Recipe Ontologies
31	*Changes in eating patterns*	Yes	_

Table VIII, *List of dangerous and nonsense scenarios in AAL. Highlighted scenarios are those on which is based the proposed Bayesian Network.*

The scenarios have been chosen depending on the types of activity that the Low-level knowledge model is able to detect. In fact, by associating the activities contained in the NTU RGB+D 60 dataset *(Tab. VI)* with all the scenarios shown in *(Tab. VIII)*, we have defined some possible *Situations (Tab. IX).* Situations may represent feelings, events, behaviours, or intentions as shown in *(Tab. V, Chapter 3.2.4.4)*.

Then, each situation has been cross-referenced against actions, sensor readings, and other situations in order to find *Anomalous Scenarios (Tab. X)*. Anomalies represent dangerous or nonsense scenarios that may happen in the environment depending on the recognizable evidence and situations. At this high level of knowledge, the system achieves a significant degree of awareness.

			Situations															
N	Code	Discrete actions and sensor's input	Feeling Hot	Feeling Cold	Get undressed	Get dressed	Getting back home	Leaving home	Washing	Preparing a meal	Standing Up	Sitting	Eating	Waiting	Searching something	Interacting	Pastime	Smth on the ground
			HO	CO	GU	GD	GB	LH	WS	PM	SU	SI	EA	WT	SS	IN	PA	SG
1	A1	drink water																
2	A2	eat meal																
3	A3	brush teeth																
4	A4	brush hair																
5	A5	drop																
6	A6	pick up																
7	A7	throw																
8	A8	sit down																
9	A9	stand up																
10	A10	clapping																
11	A11	reading																
12	A12	writing																
13	A13	tear up paper																
14	A14	put on jacket																
15	A15	take off jacket																
16	A16	put on a shoe																
17	A17	take off a shoe																
18	A18	put on glasses																
19	A19	take off glasses																
20	A20	put on a hat/cap																
21	A21	take off a hat/cap																
22	A22	cheer up																
23	A23	hand waving																
24	A24	kicking something																
25	A25	reach into pocket																
26	A26	hopping																
27	A27	jump up																
28	A28	phone call																
29	A29	play with phone/tablet																
30	A30	type on a keyboard																
31	A31	point to something																
32	A32	taking a selfie																
33	A33	check time (from watch)																
34	A34	rub two hands																
35	A35	nod head/bow																
36	A36	shake head																
37	A37	wipe face																
38	A38	salute																
39	A39	put palms together																
40	A40	cross hands in front																
49	A49	fan self																
N	Code	Sensors																
1	S1	Stove																
2	S2	Oven																
3	S3	Fridge																
4	S4	Sink																
5	S5	Door (entrance)																
6	S6	Window																
7	S7	Indoor Temperature																
8	S8	Outdoor Temperature																

Table IX, *Situations in our AAL environment.*

			Anomaly awareness																
N	Code	Discrete actions and sensor's input	Risk leaving home	Why not leaving?	Risk leaving open	Risk leaving open water	Risk leaving appliances	Eating standing up	Waiting cooking time	Risk falling	Too many activities	Unpleasant reading	Writing error	Angry	Risk leaving things on	Why window open?	Why window close?	Eating time anomaly	Cooking time anomaly
			RLU	WNL	RLW	RSW	RSA	ESU	WCT	RFA	TMA	URE	WER	ANG	LTG	WWO	WWC	ETA	CTA
5	A5	drop																	
6	A6	pick up																	
7	A7	throw																	
11	A11	reading																	
12	A12	writing																	
13	A13	tear up paper																	
24	A24	kicking something																	
N	Code	Sensors																	
4	S4	Sink																	
5	S5	Door																	
6	S6	Window																	
7	S7	Indoor Temperature																	
8	S8	Outdoor Temperature																	
	Code	Situation																	
	HO	Feeling Hot																	
	CO	Feeling Cold																	
	GU	Get undressed																	
	GD	Get dressed																	
	GB	Getting back home																	
	LH	Leaving Home																	
	WS	Washing																	
	PM	Preparing a meal																	
	SU	Standing up																	
	SI	Sitting																	
	EA	Eating																	
	WT	Waiting																	
	SS	Searching something																	
	IN	Interacting																	
	PA	Pastime																	
	SG	Smth on the ground																	
	Code	Other																	
	TT	Timetable																	

Table X, *Anomaly awareness in our AAL environment.*

In *(Fig. 38)* is shown a scheme that summarizes the *symptoms-anomalies extraction process*.

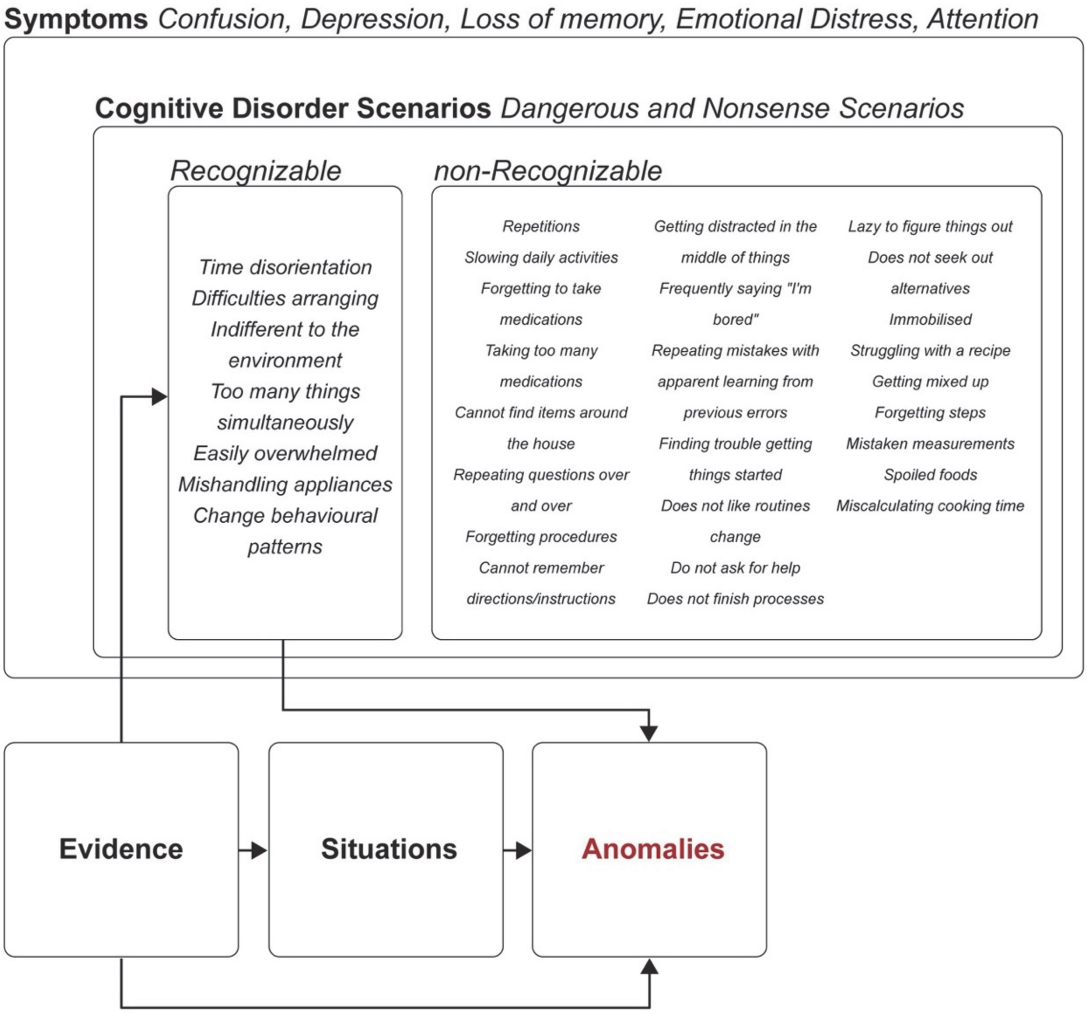


Figure 38, *Symptoms-anomalies flow. Symptoms define dangerous and nonsense scenarios. Those are distinguishable through evidence. Recognizable anomalies are those included in the list of the Dangerous and Nonsense recognizable scenarios, that may be caused by either specific situations or activities (Evidence), or by both simultaneously.*

4.3.2. Development of the networks

Following the flow indicated in *(Fig. 38)*, we have developed four main flexible, scalable, reusable and interconnected Bayesian network modules that can recognize:

- *Dangerous and Nonsense scenarios*
- *Changes in eating/cooking patterns*
- *Environmental discomfort*
- *Emergency situations*

The whole OOBN is shown in *(Fig. 39). Single modules are described in depth in next chapters.*

Firstly, nodes (evidence, situations, and anomalies) and causal dependences have been defined. Anomalies depend on both situations and evidence, while situations only depend on evidence. The input nodes, represented by dashed edges in *(Fig. 39)*, derive from three different types of sources:

1. NNs action output scores (percentages).
2. Sensor readings (e.g., stove on or off).
3. Interactions with the user (e.g., “I’m feeling cold”).

Output nodes, represented by solid edges in *(Fig. 39)*, will send messages to the Automation HUB when required (e.g., “Turn Heating On” or “Interact with the user in some way” or “Remind the user something” or “Suggest something”). Nevertheless, there are some output nodes that allow connection between different modules as well as fulfilling their role as messengers, i.e., *PreparingMeal*, *EatingMeal*, *ObjectAnomalies*, *FeelingCold*.

Then, Conditional Probability tables (CPTs) have been filled eliciting the author’s knowledge. Marginal probabilities regarding input nodes are unspecified since we have not trained the network in a real environment. Thus, their tables have been filled with total uncertainty values (e.g., 50% probability for binary-type nodes). However, the prior probability of the nodes representing actions could be updated with the action output scores coming from the NNs, to feed the BNs with consistent initial values. After preliminary tests, nodes, dependencies, and CPTs have been fixed due to conflicts in some aggregations.

The result is an effective High-level knowledge model, able to reason on low-level information to yield likelihoods of pre-defined anomalies that could happen in an AAL environment. Although the model is built to perform deductive reasoning to fulfil his role (a top-down reasoning, from causes to consequences), it would allow abductive reasoning as well. In the following chapters it is shown only the deductive approach. Abductive approach could be explored thoroughly in future developments. Besides, the networks are built to support the user only when required. Interacting frequently with the user may let him or her be reluctant, or even sad that he or she could not keep independence.

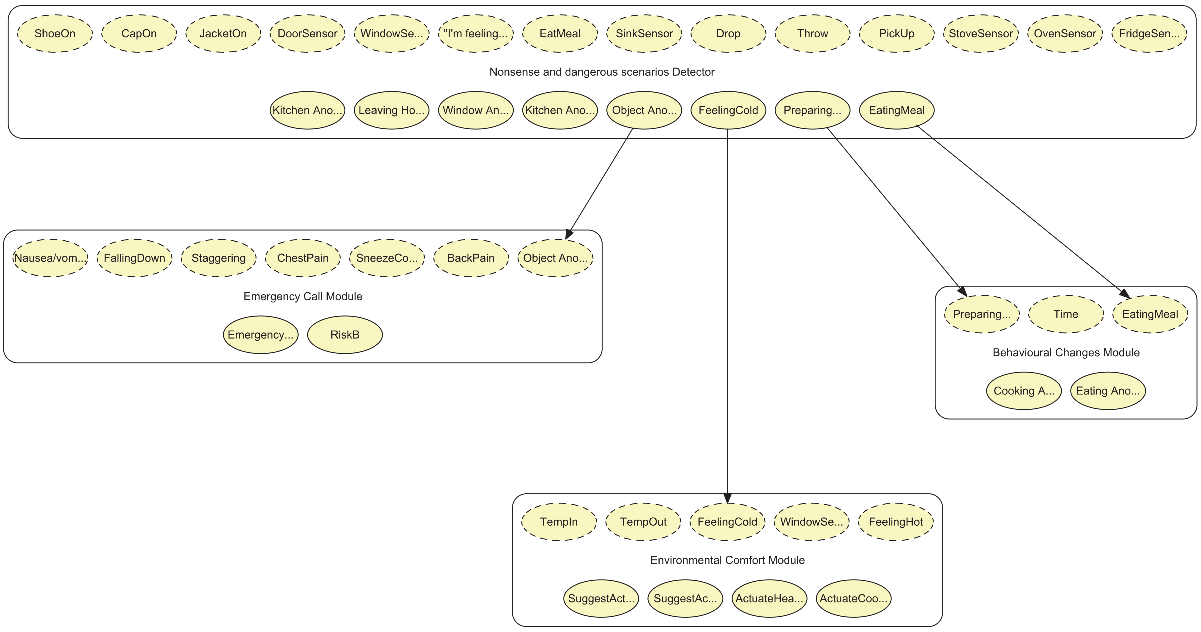


Figure 39, *Object-Oriented Bayesian Network that represent the High-level knowledge model of our system.*

4.3.2.1 Nonsense and dangerous scenarios detector

The first module of our Object-Oriented Bayesian Network consists in a unit that can recognize nonsense and dangerous scenarios. Those are clearly limited in number because of both the type of actions that the low-level knowledge model can detect and the sensors that we have deployed in the environment. Nevertheless, this module could be updated by adding additional actions and sensors to recognize a larger number of situations and anomalies. The module network is illustrated in *(Fig. 40)*. The nodes CPTs are shown in *(Tab. XI)*.

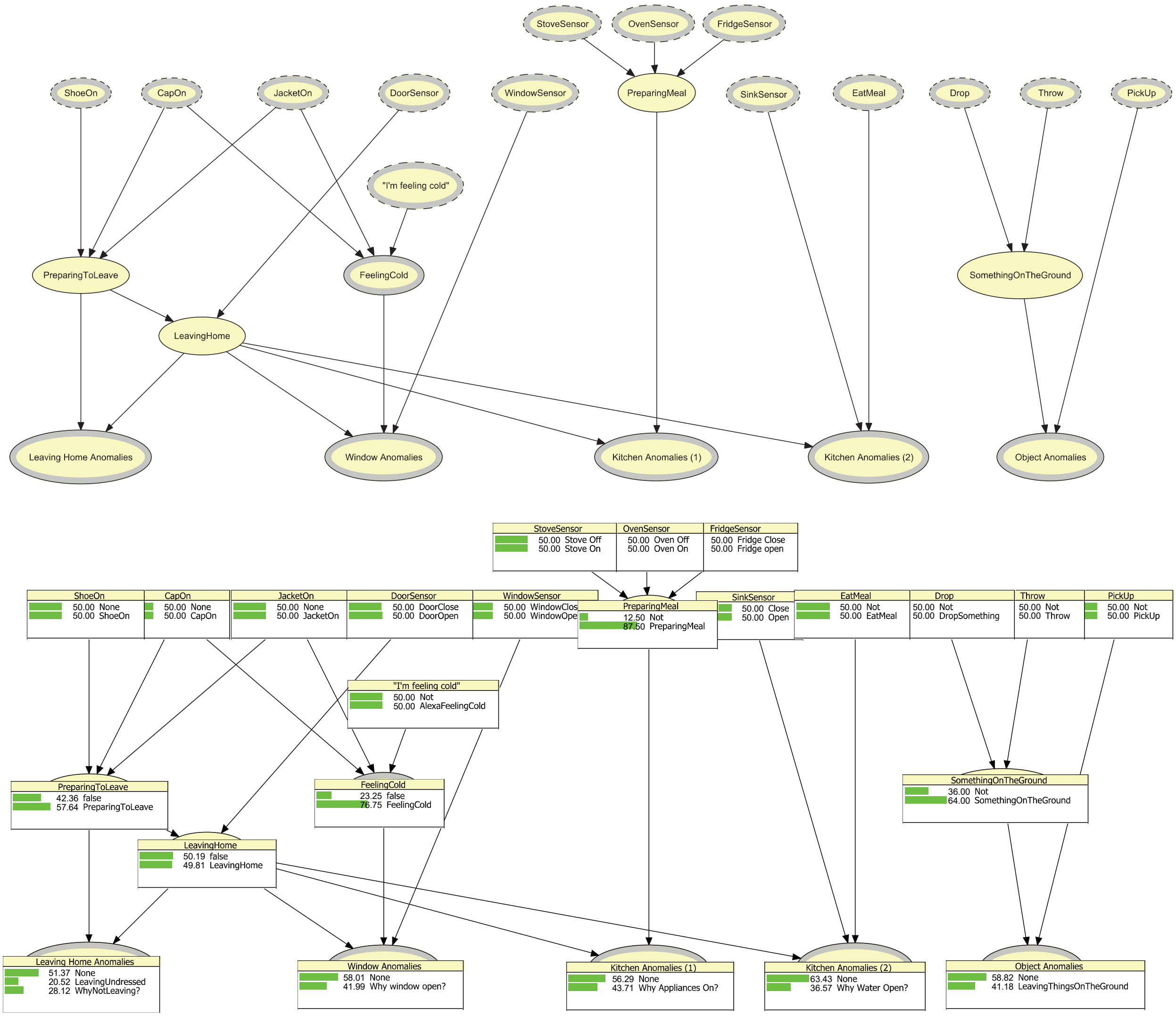


Figure 40, *Network of the Nonsense and dangerous scenarios detector, without (top) and with (down) belief monitor windows. No evidence has been provided yet.*

Object Anomalies				
PickUp	Not		PickUp	
SomethingOn TheGround	Not	SomethingOn TheGround	Not	SomethingOn TheGround
None	0.5	0.01	0.99	0.99
LeavingThings OnTheGround	0.5	0.99	0.01	0.01

SomethingOnTheGround				
Throw	Not		Throw	
Drop	Not	DropSomething	Not	DropSomething
None	0.99	0.2	0.2	0.05
LeavingThings OnTheGround	0.01	0.8	0.8	0.95

Kitchen Anomalies (2)								
LeavingHome	False				LeavingHome			
EatMeal	Not		EatMeal		Not		EatMeal	
SinkSensor	Close	Open	Close	Open	Close	Open	Close	Open
None	0.99	0.99	0.99	0.1	0.99	0.01	0.99	0.01
Why Water Open?	0.01	0.01	0.01	0.9	0.01	0.99	0.01	0.99

PreparingToLeave								
ShoeOn	False				LeavingHome			
CapOn	Not		CapOn		Not		CapOn	
JacketOn	Not	JacketOn	Not	JacketOn	Not	JacketOn	Not	JacketOn
False	0.9	0.8	0.6	0.5	0.25	0.1	0.1	0.05
PreparingTo Leave	0.01	0.2	0.4	0.5	0.75	0.9	0.9	0.95

FeelingCold								
"I'm Feeling Cold"	Not				DialogSystem interaction "I'm feeling cold"			
CapOn	Not		CapOn		Not		CapOn	
JacketOn	Not	JacketOn	Not	JacketOn	Not	JacketOn	Not	JacketOn
False	0.9	0.4	0.25	0.1	0.1	0.05	0.05	0.01
FeelingCold	0.1	0.6	0.75	0.9	0.9	0.95	0.95	0.99

Window Anomalies								
WindowSensor	WindowClose				WindowOpen			
LeavingHome	false		LeavingHome		false		LeavingHome	
FeelingCold	Not	FeelingCold	Not	FeelingCold	Not	FeelingCold	Not	FeelingCold
None	0.99	0.99	0.99	0.99	0.99	0.1	0.1	0.01
Why Window Open?	0.01	0.01	0.01	0.01	0.01	0.9	0.9	0.99

LeavingHome				
DoorSensor	DoorClose		DoorOpen	
PreparingTo Leave	false	PreparingTo Leave	false	PreparingTo Leave
None	0.95	0.95	0.1	0.02
LeavingHome	0.05	0.05	0.9	0.98

LeavingHome Anomalies				
LeavingHome	false		LeavingHome	
PreparingTo Leave	false	PreparingTo Leave	false	PreparingTo Leave
None	0.98	0.01	0.01	0.98
Leaving Undressed	0.01	0.01	0.98	0.01
Why Not Leaving?	0.01	0.98	0.01	0.01

Kitchen Anomalies (1)				
LeavingHome	false		LeavingHome	
PreparingMeal	Not	PreparingMeal	Not	PreparingMeal
None	0.95	0.95	0.1	0.02
Why Appliances On?	0.05	0.05	0.9	0.98

Tab. XI, *CPTs of the nodes in the Nonsense and dangerous scenarios detector.*

4.3.2.2 Behavioural change detector

The second module of our Object-Oriented Bayesian Network consists in a unit that can recognize behavioural change regarding both the cooking and eating activities. Since patients suffering from cognitive disorders are subject to changes in their behaviours, this unit aims at supporting the user in maintaining a balanced life in terms of regularity of the meals. This module could be updated by adding additional types of behaviours that may change due to diseases, but the structure will remain the same. The module network is illustrated in *(Fig. 41)*. The nodes CPTs are shown in *(Tab. XII)*.

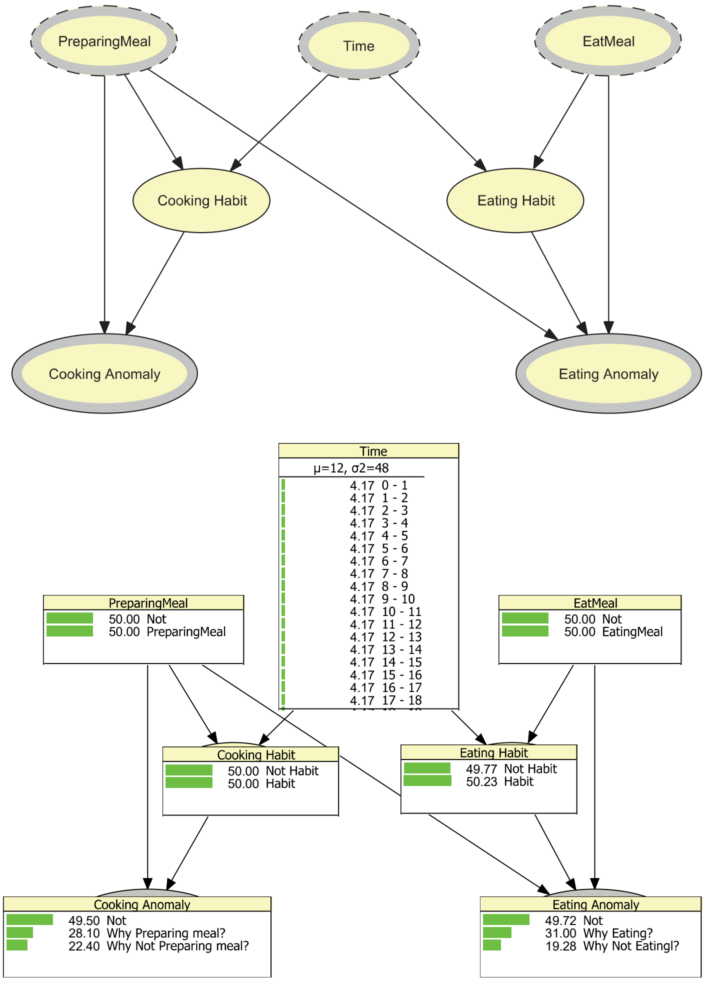


Figure 41, *Network of the Behavioural Changes detector, without (top) and with (down) belief monitor windows. No evidence has been provided yet.*

Eating Anomaly

PreparingMeal	Not				PreparingMeal			
EatMeal	Not		EatingMeal		Not		EatingMeal	
Habit_1	Not Habit	Habit	Not Habit	Habit	Not Habit	Habit	Not Habit	Habit
Not	0.01	0.98	0.01	0.98	0.01	0.98	0.01	0.98
Why Eating?	0.01	0.01	0.98	0.01	0.5	0.01	0.98	0.01
Why Not Eating?	0.98	0.01	0.01	0.01	0.49	0.01	0.01	0.01

Eating Habit

EatMeal	Not									
Time	0-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10
Not Habit	0.01	0.01	0.01	0.01	0.05	0.1	0.4	0.7	0.9	0.95
Habit	0.99	0.99	0.99	0.99	0.95	0.9	0.6	0.3	0.1	0.05

EatMeal	Not									
Time	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20
Not Habit	0.5	0.5	0.7	0.95	0.98	0.5	0.5	0.5	0.5	0.8
Habit	0.5	0.5	0.3	0.05	0.02	0.5	0.5	0.5	0.5	0.2

EatMeal	Not				EatingMeal					
Time	20-21	21-22	22-23	23-24	0-1	1-2	2-3	3-4	4-5	5-6
Not Habit	0.95	0.98	0.5	0.1	0.99	0.99	0.99	0.99	0.95	0.9
Habit	0.05	0.02	0.5	0.9	0.01	0.01	0.01	0.01	0.05	0.1

EatMeal	EatingMeal									
Time	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16
Not Habit	0.6	0.3	0.1	0.05	0.5	0.5	0.3	0.05	0.8	0.5
Habit	0.4	0.7	0.9	0.95	0.5	0.5	0.7	0.95	0.2	0.5

EatMeal	EatingMeal							
Time	16-17	17-18	18-19	19-20	20-21	21-22	22-23	23-24
Not Habit	0.5	0.5	0.5	0.2	0.05	0.02	0.5	0.01
Habit	0.5	0.5	0.5	0.8	0.95	0.98	0.5	0.99

Cooking Anomaly

PreparingMeal	Not		PreparingMeal	
Habit	Not Habit	Habit	Not Habit	Habit
Not	0.01	0.98	0.01	0.98
Why Preparing Meal?	0.01	0.01	0.98	0.01
Why Not Preparing Meal?	0.98	0.01	0.01	0.01

Cooking Habit

PreparingMeal	Not									
Time	0-1	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10
Not Habit	0.01	0.01	0.01	0.01	0.05	0.1	0.4	0.7	0.9	0.95
Habit	0.99	0.99	0.99	0.99	0.95	0.9	0.6	0.3	0.1	0.05

PreparingMeal	Not									
Time	10-11	11-12	12-13	13-14	14-15	15-16	16-17	17-18	18-19	19-20
Not Habit	0.5	0.5	0.7	0.95	0.2	0.5	0.5	0.5	0.7	0.9
Habit	0.5	0.5	0.3	0.05	0.8	0.5	0.5	0.5	0.3	0.1

PreparingMeal	Not				PreparingMeal					
Time	20-21	21-22	22-23	23-24	0-1	1-2	2-3	3-4	4-5	5-6
Not Habit	0.95	0.3	0.2	0.05	0.99	0.99	0.99	0.99	0.95	0.9
Habit	0.05	0.7	0.8	0.95	0.01	0.01	0.01	0.01	0.05	0.1

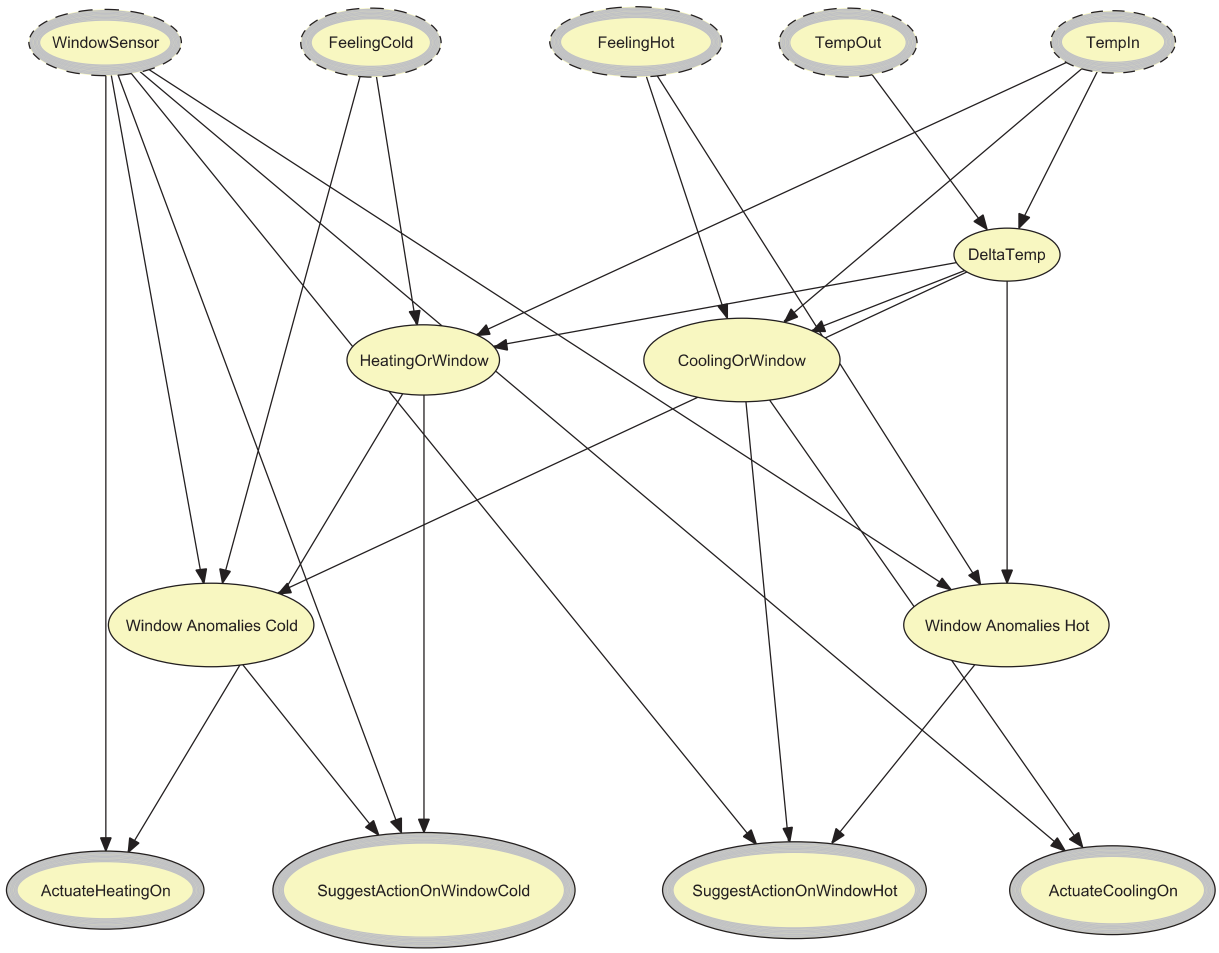
PreparingMeal	PreparingMeal									
Time	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15	15-16
Not Habit	0.6	0.3	0.1	0.05	0.5	0.5	0.3	0.05	0.8	0.5
Habit	0.4	0.7	0.9	0.95	0.5	0.5	0.7	0.95	0.2	0.5

PreparingMeal	PreparingMeal							
Time	16-17	17-18	18-19	19-20	20-21	21-22	22-23	23-24
Not Habit	0.5	0.5	0.3	0.1	0.05	0.7	0.8	0.95
Habit	0.5	0.5	0.7	0.9	0.95	0.3	0.2	0.05

Tab. XII, *CPTs of the nodes in the Behavioural Changes detector.*

4.3.2.3 Environmental comfort module

The third module of our Object-Oriented Bayesian Network consists in a unit that can recognize environmental discomfort. Since patients suffering from cognitive disorders may remain indifferent to the environment, due to symptoms such as confusion and so on, he or she would not recognize situation of discomfort and waste of energy. Hence, this module leads to an environmental awareness. The module network is illustrated in *(Fig. 42)*. The nodes CPTs are shown in *(Tab. XIII)*.



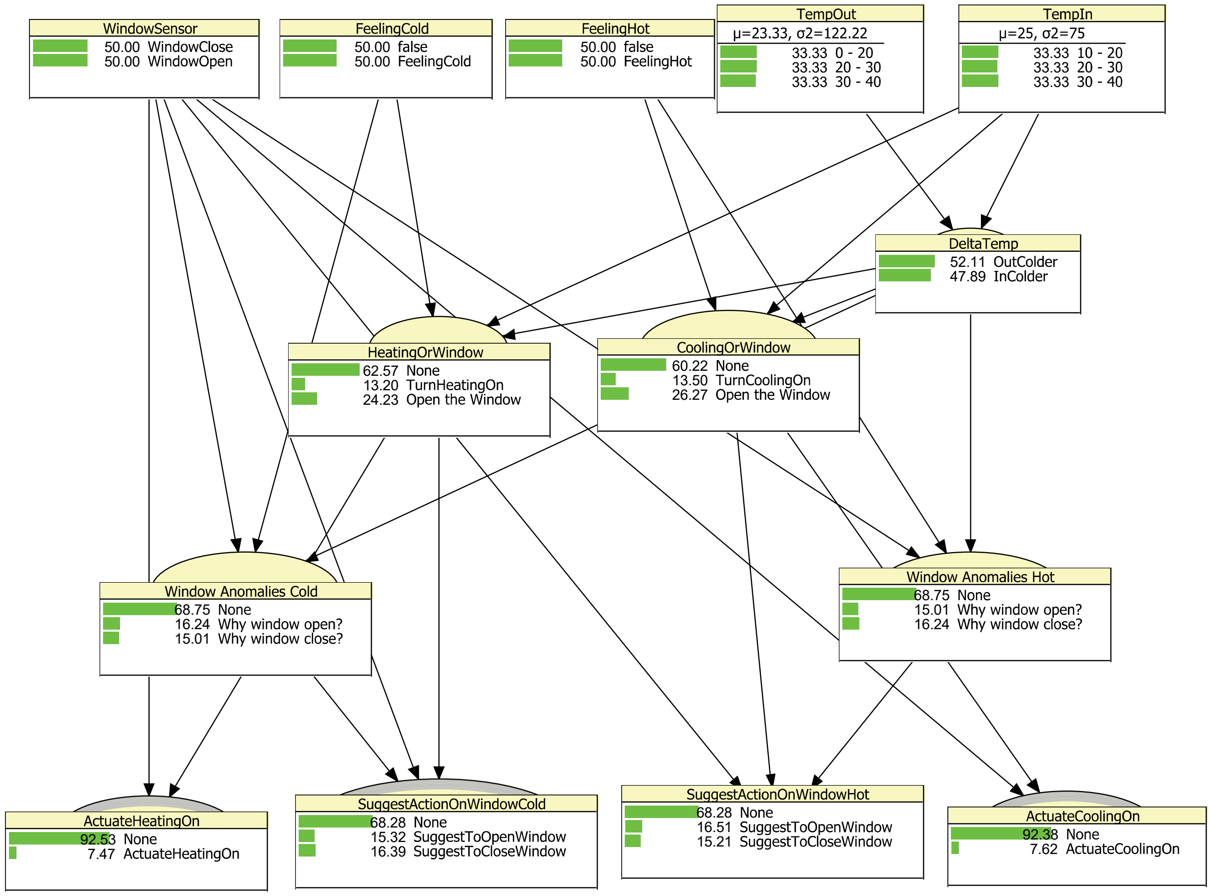


Figure 42, *Network of the Environmental Comfort module, without (previous page) and with (this page) belief monitor windows. No evidence has been provided yet.*

DeltaTemp									
TempIn	10-20			20-30			30-40		
TempOut	0-20	20-30	30-40	0-20	20-30	30-40	0-20	20-30	30-40
OutColder	0.95	0.05	0.01	0.99	0.5	0.05	0.99	0.95	0.2
InColder	0.05	0.95	0.99	0.01	0.5	0.95	0.01	0.05	0.8

HeatingOrWindow												
TempIn	10-20				20-30				30-40			
DeltaTemp	OutColder		InColder		OutColder		InColder		OutColder		InColder	
FeelingCold	false	FeelingCold	false	FeelingCold	false	FeelingCold	false	FeelingCold	false	FeelingCold	false	FeelingCold
None	0.98	0.01	0.98	0.01	0.98	0.19	0.98	0.01	0.98	0.98	0.98	0.01
TurnHeatingOn	0.01	0.98	0.01	0.01	0.01	0.8	0.01	0.01	0.01	0.01	0.01	0.01
Open the Window	0.01	0.01	0.01	0.98	0.01	0.01	0.01	0.98	0.01	0.01	0.01	0.98

CoolingOrWindow												
TempIn	10-20				20-30				30-40			
DeltaTemp	OutColder		InColder		OutColder		InColder		OutColder		InColder	
FeelingHot	false	FeelingHot	false	FeelingHot	false	FeelingHot	false	FeelingHot	false	FeelingHot	false	FeelingHot
None	0.98	0.01	0.98	0.98	0.98	0.01	0.98	0.01	0.98	0.01	0.98	0.01
TurnCoolingOn	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.98	0.01	0.01	0.01	0.98
Open the Window	0.01	0.98	0.01	0.01	0.01	0.98	0.01	0.01	0.01	0.98	0.01	0.01

Window Anomalies Cold								
FeelingCold	false				FeelingCold			
DeltaTemp	OutColder		InColder		OutColder		InColder	
WindowSensor	WindowClose	WindowOpen	WindowClose	WindowOpen	WindowClose	WindowOpen	WindowClose	WindowOpen
None	0.98	0.78	0.78	0.98	0.98	0.01	0.01	0.98
Why Window Open	0.01	0.21	0.01	0.01	0.01	0.98	0.01	0.01
Why Window Close	0.01	0.01	0.21	0.01	0.01	0.01	0.98	0.01

Window Anomalies Hot								
FeelingHot	false				FeelingHot			
DeltaTemp	OutColder		InColder		OutColder		InColder	
WindowSensor	WindowClose	WindowOpen	WindowClose	WindowOpen	WindowClose	WindowOpen	WindowClose	WindowOpen
None	0.78	0.98	0.98	0.78	0.01	0.98	0.98	0.01
Why Window Open	0.01	0.01	0.01	0.21	0.01	0.01	0.01	0.98
Why Window Close	0.21	0.01	0.01	0.01	0.98	0.01	0.01	0.01

Suggest Action On Window (Cold)												
Window Anomalies Cold	None						Why Window open?					
WindowSensor	WindowClose			WindowOpen			WindowClose			WindowOpen		
HeatingOr Window	None	TurnHeatingOn	Open the window	None	TurnHeatingOn	Open the window	None	TurnHeatingOn	Open the window	None	TurnHeatingOn	Open the window
None	0.98	0.98	0.01	0.98	0.01	0.98	0.98	0.98	0.98	0.01	0.01	0.98
Suggest to Open the Window	0.01	0.01	0.98	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Suggest to Close the Window	0.01	0.01	0.01	0.01	0.98	0.01	0.01	0.01	0.01	0.98	0.98	0.01

Window Anomalies Cold	Why Window close?					
WindowSensor	WindowClose			WindowOpen		
HeatingOr Window	None	TurnHeatingOn	Open the window	None	TurnHeatingOn	Open the window
None	0.01	0.98	0.01	0.98	0.98	0.98
Suggest to Open the Window	0.98	0.01	0.98	0.01	0.01	0.01
Suggest to Close the Window	0.01	0.01	0.01	0.01	0.01	0.01

Suggest Action On Window (Hot)												
Window Anomalies Hot	None						Why Window open?					
WindowSensor	WindowClose			WindowOpen			WindowClose			WindowOpen		
CoolingOr Window	None	TurnHeatingOn	Open the window	None	TurnHeatingOn	Open the window	None	TurnHeatingOn	Open the window	None	TurnHeatingOn	Open the window
None	0.98	0.98	0.01	0.98	0.01	0.98	0.98	0.98	0.98	0.01	0.01	0.98
Suggest to Open the Window	0.01	0.01	0.98	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Suggest to Close the Window	0.01	0.01	0.01	0.01	0.98	0.01	0.01	0.01	0.01	0.98	0.98	0.01

Window Anomalies Cold	Why Window close?					
WindowSensor	WindowClose			WindowOpen		
CoolingOr Window	None	TurnHeatingOn	Open the window	None	TurnHeatingOn	Open the window
None	0.01	0.98	0.01	0.98	0.98	0.98
Suggest to Open the Window	0.98	0.01	0.98	0.01	0.01	0.01
Suggest to Close the Window	0.01	0.01	0.01	0.01	0.01	0.01

Actuate Heating On						
WindowSensor	WindowClose			WindowOpen		
HeatingOr Window	None	TurnHeatingOn	OpenTheWindow	None	TurnHeatingOn	OpenTheWindow
None	0.99	0.01	0.99	0.99	0.99	0.99
Actuate Heating On	0.01	0.99	0.01	0.01	0.01	0.01

Actuate Cooling On						
WindowSensor	WindowClose			WindowOpen		
CoolingOr Window	None	TurnCoolingOn	OpenTheWindow	None	TurnCoolingOn	OpenTheWindow
None	0.99	0.01	0.99	0.99	0.99	0.99
Actuate Cooling On	0.01	0.99	0.01	0.01	0.01	0.01

Tab. XIII, *CPTs of the nodes in the Environmental comfort module.*

4.3.2.3 Emergency module

The fourth module of our Object-Oriented Bayesian Network consists in a simple unit that has the only purpose of triggering an emergency call in some specific cases. This module is based on some of the recognizable actions/activities of the NTU RGB+d60 dataset, namely, Sneeze/cough, Staggering, Falling down, headache, chest pain, back pain, neck pain, and nausea/vomiting. Some of those involves higher level of risks (falling down, nausea/vomiting, staggering), while others seem less dangerous. Although other Machine Learning tools, such as the Decision Tree, are definitely more suitable to this role than BNs, we have formalized this concept with the same tool to be consistent with the proposed system architecture. Furthermore, the network could be improved by eliciting expert knowledge of medicine. The module network is illustrated in *(Fig. 43)*. The nodes CPTs have been omitted because are not relevant in this module (there are only binary-type combinations).

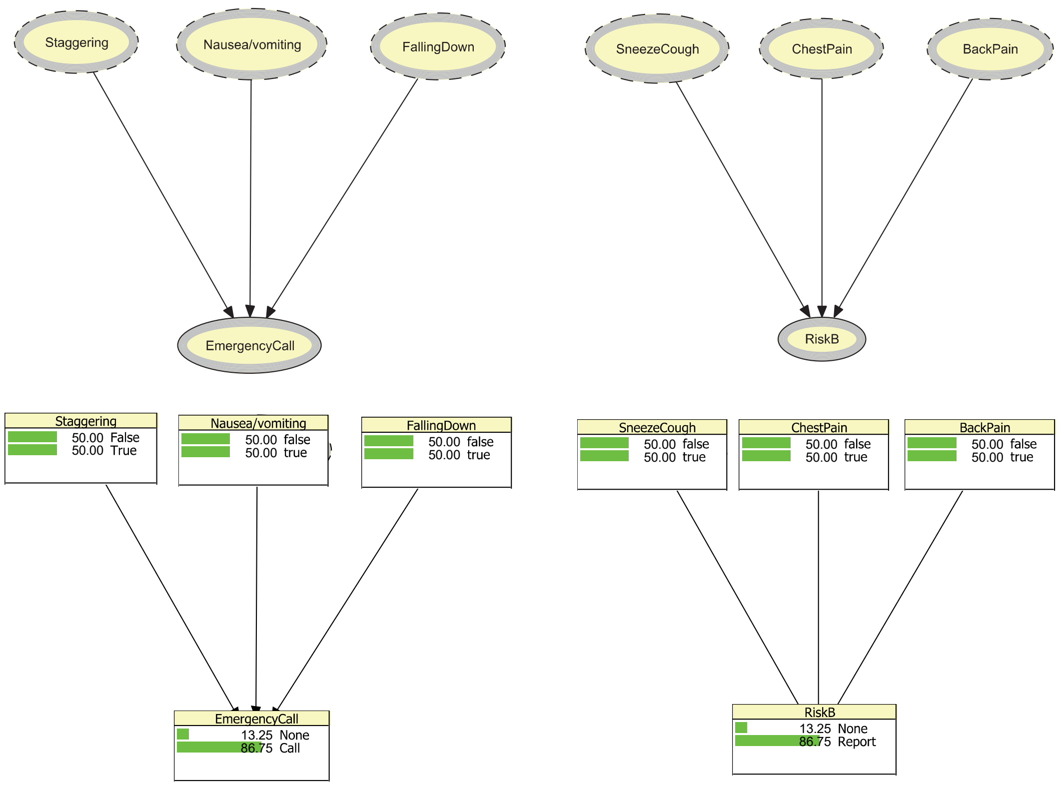


Figure 43, *Networks of the Emergency module, without (top) and with (down) belief monitor windows. No evidence has been provided yet.*

4.4 Home automation and Dialog System

Interaction between the user and the system is handled by a Visual Programming editor, i.e., Node-Red. This platform can bridge the gap between the Object-Oriented Bayesian networks, and the possible services that the system can provide to the user. Since it is responsible for communications among different components of the DT system, this module could be defined as a *Broker of Messages*. Applications offering the desired services can be integrated into the platform through the use of available palette of nodes. In our case, we integrate in our editor services that rely on Machine Learning methods to establish a Dialog System whereby bidirectional interactions between the user and the cognitive layer of the building can be performed *(Fig. 44)*.

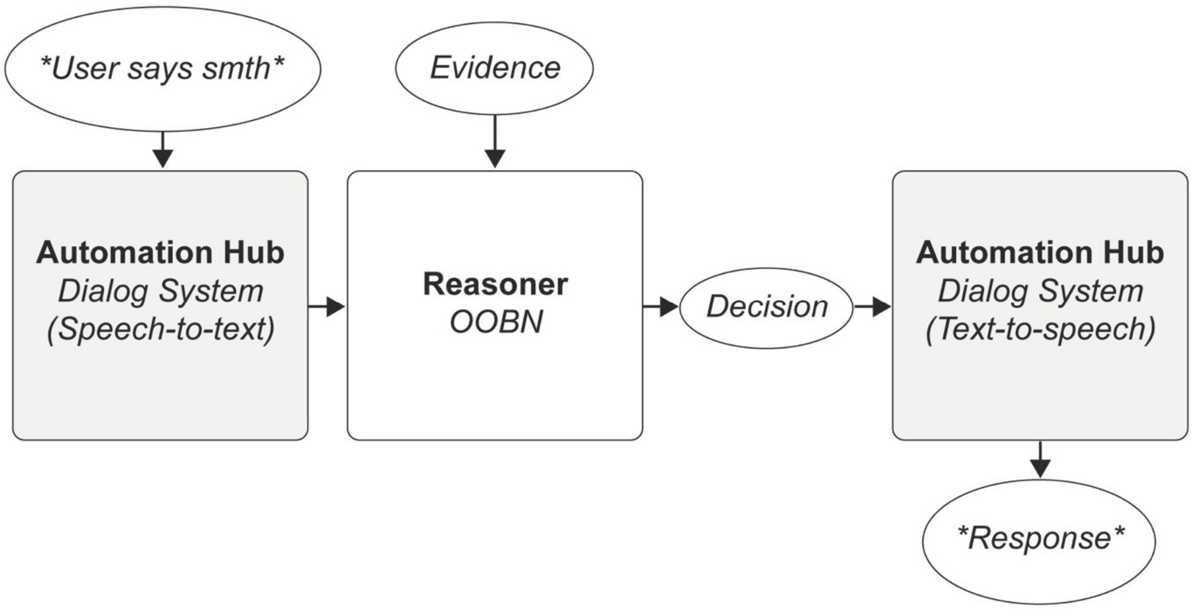


Figure 44, Flowchart *of interactions between the Reasoner and the Automation HUB.*

For what concerns the speech-to-text process, most of the Dialog Systems require a *Hot Phrase* to trigger an interaction, e.g., Google Assistant, Alexa, and so on. Examples of Hot Phrases are “Hey Google” and “Alexa”, with regard to the two Dialog System models mentioned above. Unfortunately, in our scenario the user may not be able to pronounce the Hot Phrase. For these reasons, we have moved on to other services, namely those provided by IBM, that do not need any kind of foreword. Hence, we extend the palette of nodes by installing those that act as bridges to the *IBM Watson*

Services[30]. Particularly, we exploit both the *Speech-To-Text (STT)* and *Text-To-Speech (TTS)* Machine Learning services. First, a microphone input node is deployed in the editor and wired to the STT node. The STT node converts the human voice into the written word. This service uses machine intelligence to combine information about grammar and language structure with knowledge of the composition of the audio signal to generate a more accurate transcription. It is then wired to an output node that is able to display the transcription in the debug tab. This transcription could be linked to the Bayesian Network API in order to enter the OOBN modules developed in this research. Then, an *inject* node is deployed to represent a message deriving from a decision taken by the OOBNs as a consequence of some evidence. This node is wired to a TTS node, that understands text and natural language to generate synthesized audio output complete with appropriate cadence and intonation, and finally to a node that allows an audio to be played in the browser. The figure below *(Fig. 45)* shows the flows developed into the Node-Red editor.

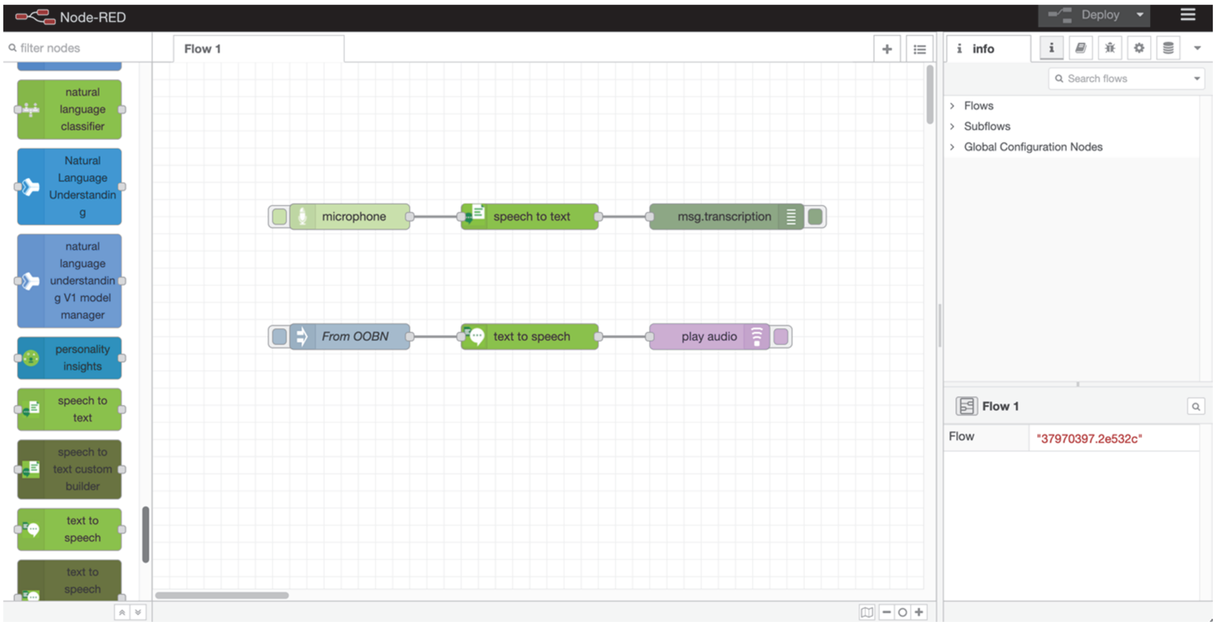


Figure 45, *Flows establishing the Dialog System.*

30 https://www.ibm.com/it-it/watson/products-services

CHAPTER V

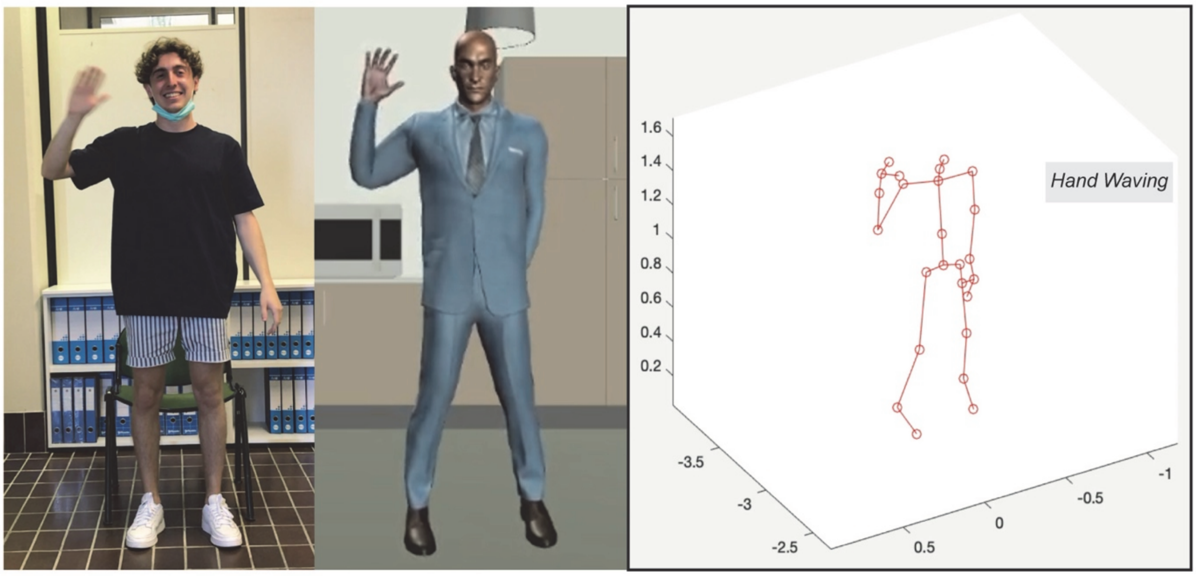
RESULTS

This chapter is divided into three parts showing results for each of the main features of the proposed system:

- *Activity Recognition (5.1)*
- *Scenario Awareness (5.2)*
- *Bidirectional User-System Interaction (5.3)*

5.1 Activity Recognition

Here are shown the results of the Activity Recognition model integrated in our DT. The validation and accuracy of the neural network model has already been provided by the developers themselves. For this reason, we test three types of action only (i.e., *hand waving*, *sit down*, *stand up*) to validate the implementation of the MS-G3D model within our environment *(Fig. 46)*.



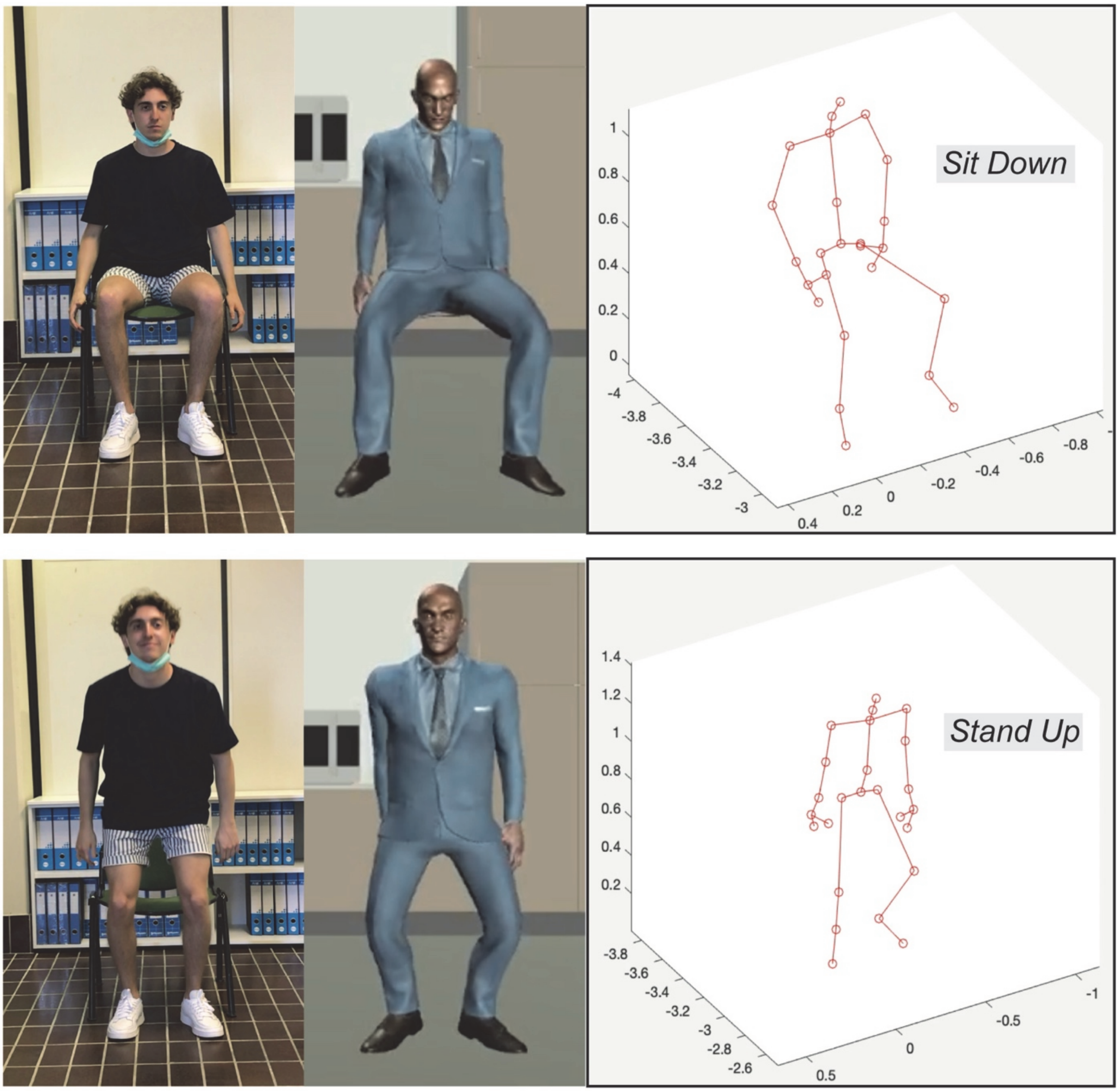


Figure 46, *Results of the Activity Recognition model.*

5.2 Scenario Awareness

To evaluate the effectiveness of the Object-Oriented Bayesian Networks developed in this work, we proceed by setting up possible combinations of evidences in the input nodes (activities, interactions, time, and so on). This helps us to demonstrate the consistency of the decisions taken by the single OOBN modules. As we can see from the images in the following paragraphs, the effectiveness of our Bayesian networks is proven. As a matter of fact, following the provision of evidences, the expected consequences achieve high percentage values. It means that predictable anomalies within the scenario are fully recognized.

5.2.1 Nonsense and dangerous scenarios detector

Some evidence has been manually provided to demonstrate the consistency of the decisions taken by this module in different scenarios *(Fig. 47 to 56)*.

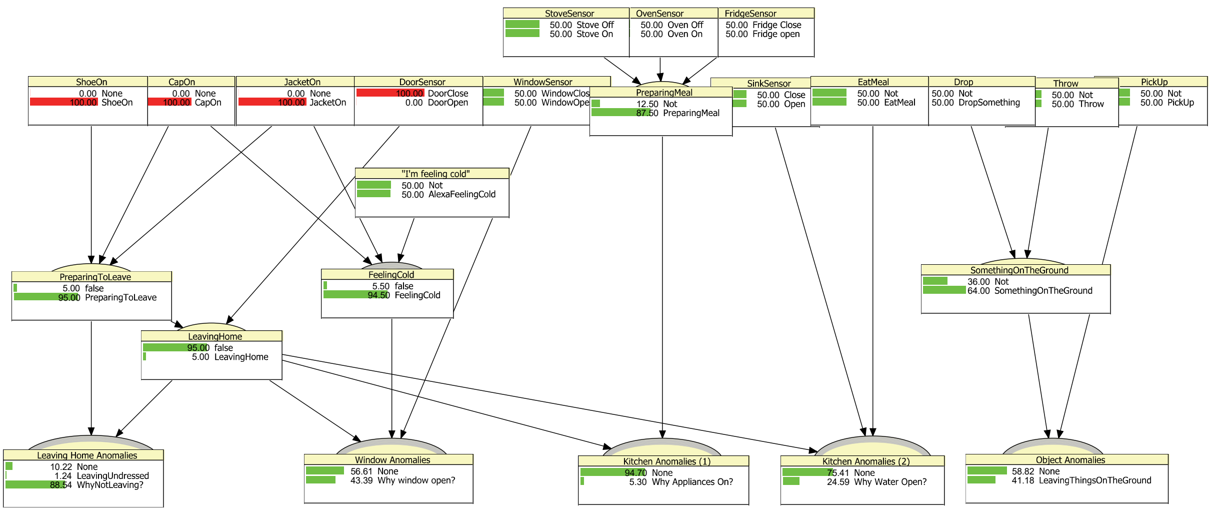


Figure 47, *The user has prepared to leave, but remains at home. The system will ask for reasons why he or she has dressed.*

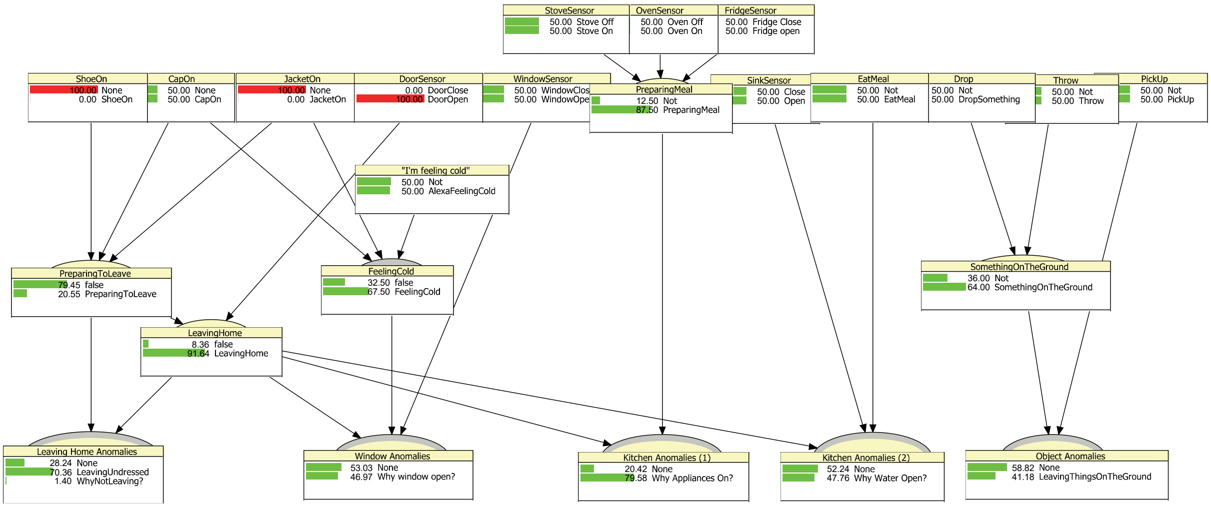


Figure 48, *The user is leaving home without shoes and jacket. He or she is undressed. The system will alert the user, reminding him or her to get dressed before leaving.*

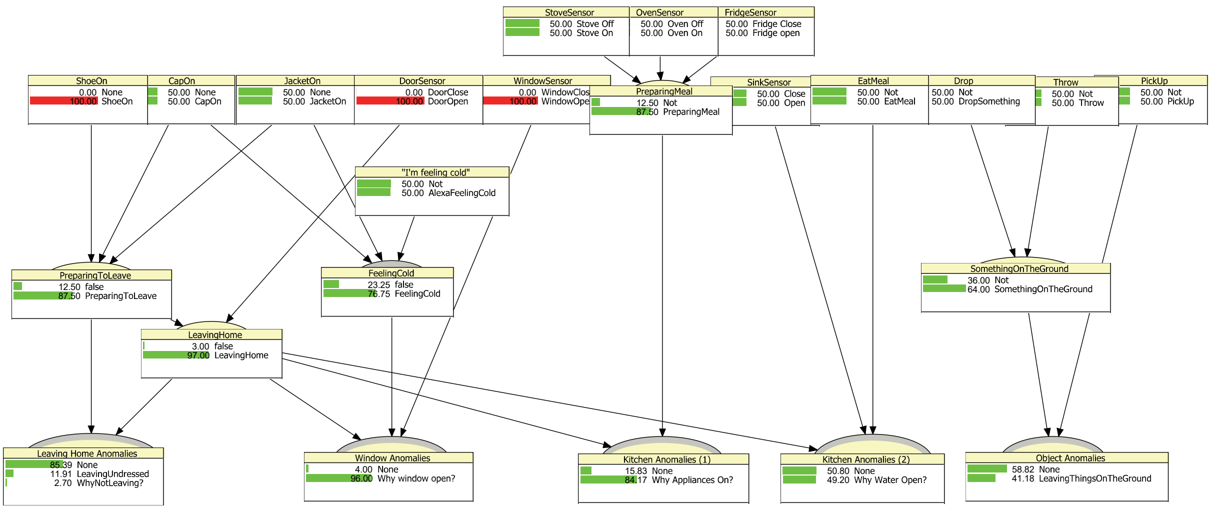


Figure 49, *The user is leaving home without closing the window. The system will alert the user, reminding him or her to close the window before leaving.*

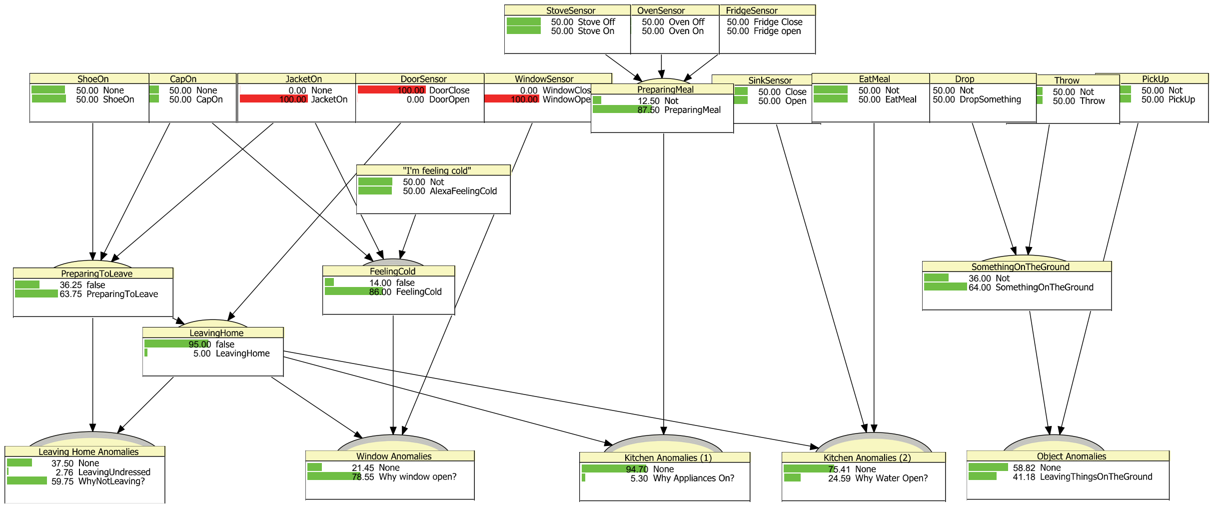


Figure 50, *The user is feeling cold but still keep the window open. The system will suggest closing the window.*

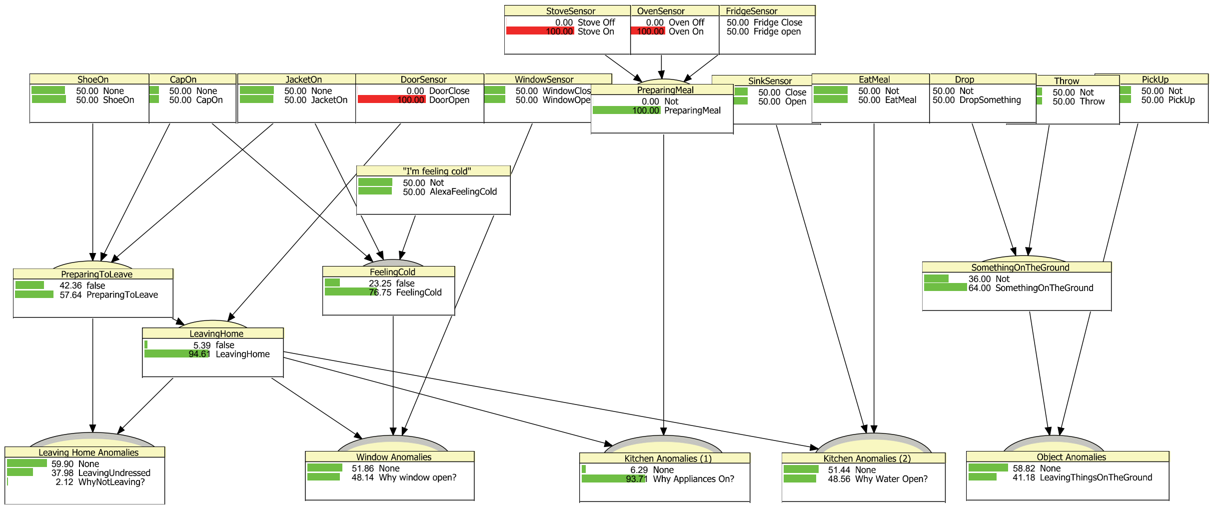


Figure 51, *The user is leaving home, forgetting some appliances turned on. The system will suggest turning off the appliances before leaving.*

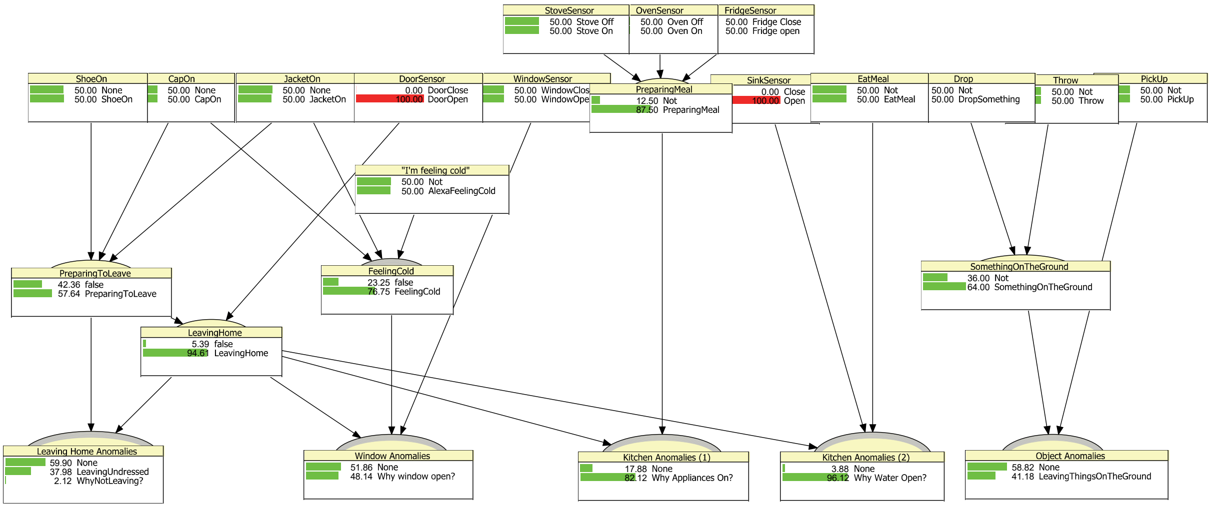


Figure 52, *The user is leaving home, forgetting to close the tap. The system will suggest closing the tap before leaving.*

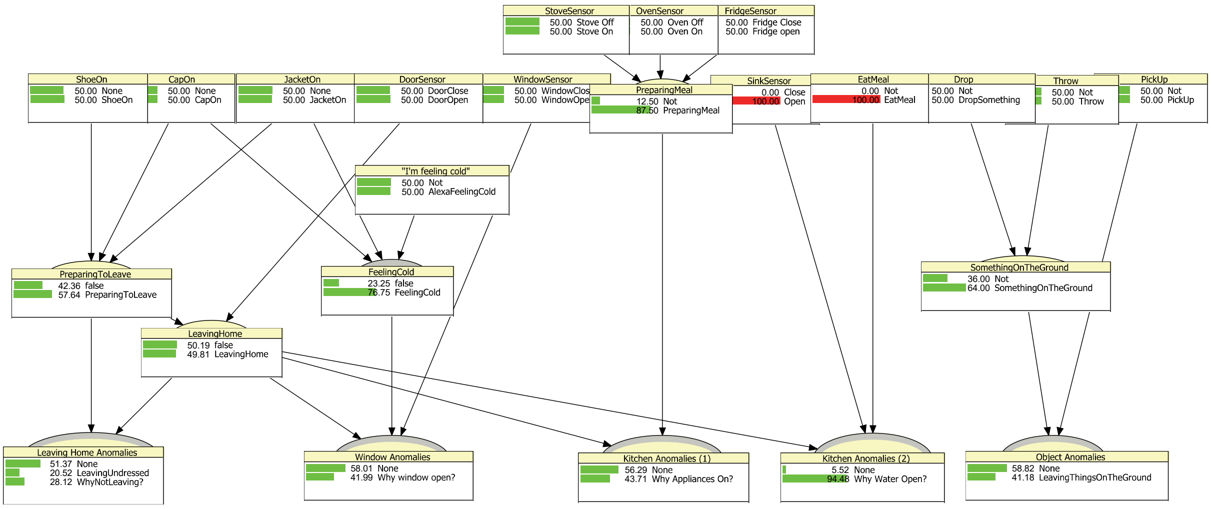


Figure 53, *The user is eating, forgetting to close the tap. The system will suggest closing the tap to save water.*

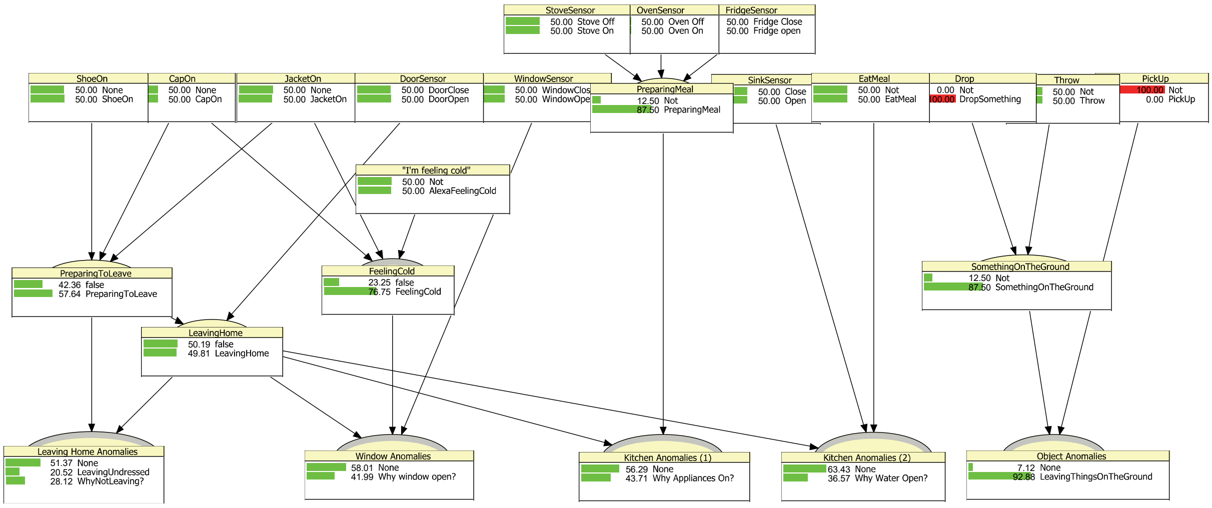


Figure 54, *Something has dropped and the user have not picked it up yet. The system will suggest picking it up to avoid falls.*

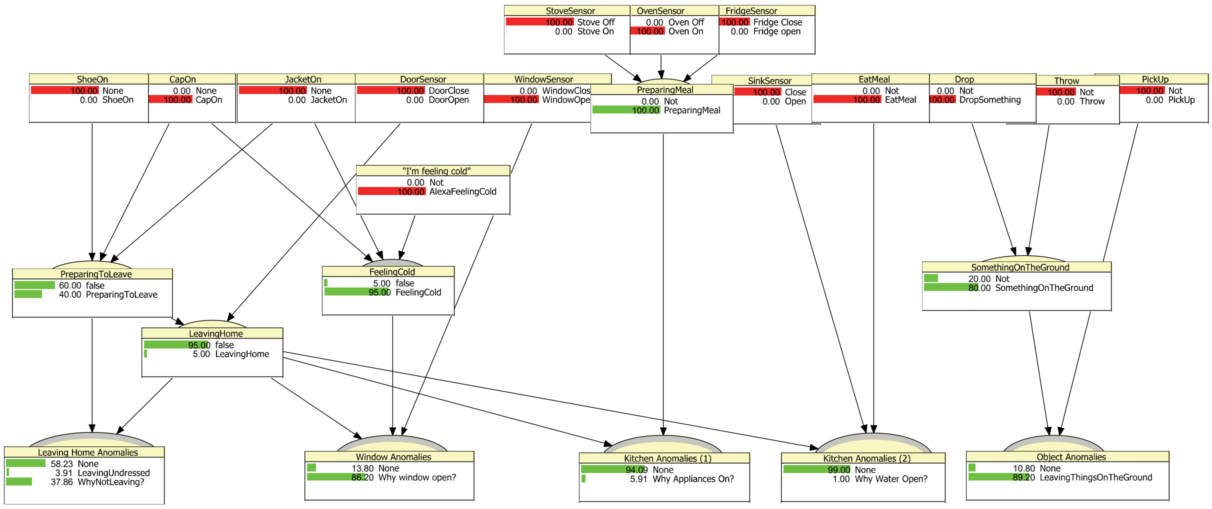


Figure 55, *Simultaneous anomalies scenario: the user has dropped something while eating. The oven is turned on. He or she is feeling cold, but the window is still open.*

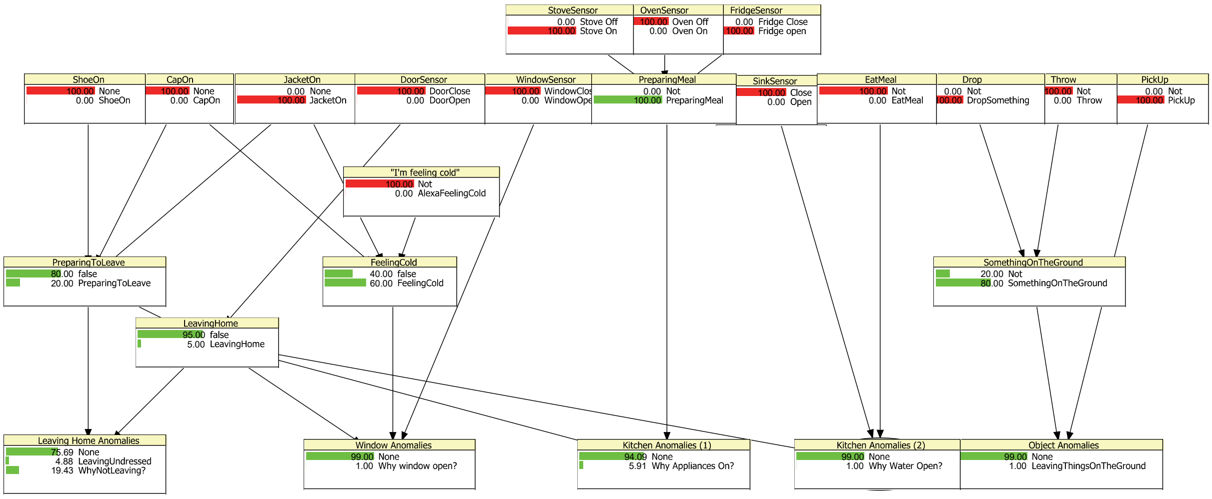


Figure 56, *Simultaneous normal situation scenario: the user has dropped something putting a jacket on, while preparing a meal. He or she has picked it up. No wasteful, nonsense or risk behaviour is detected indeed.*

5.2.2 Behavioural change detector

Some evidence has been manually provided to demonstrate the consistency of the decisions taken by this module in different scenarios *(Fig. 57 to 62)*.

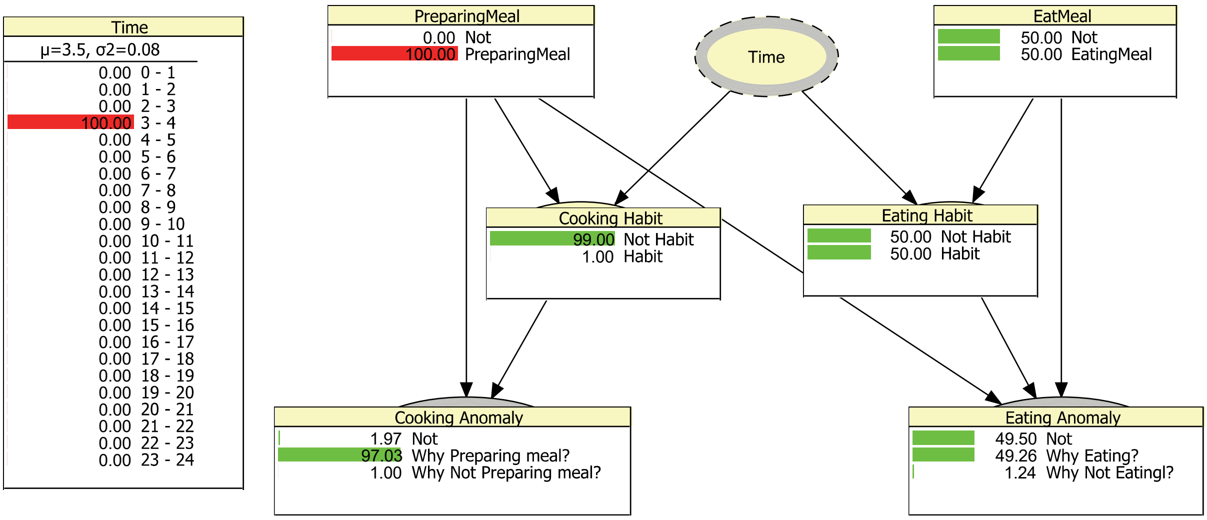


Figure 57, *The user starts preparing a meal during the night (time range between 3:00am and 4:00am).*

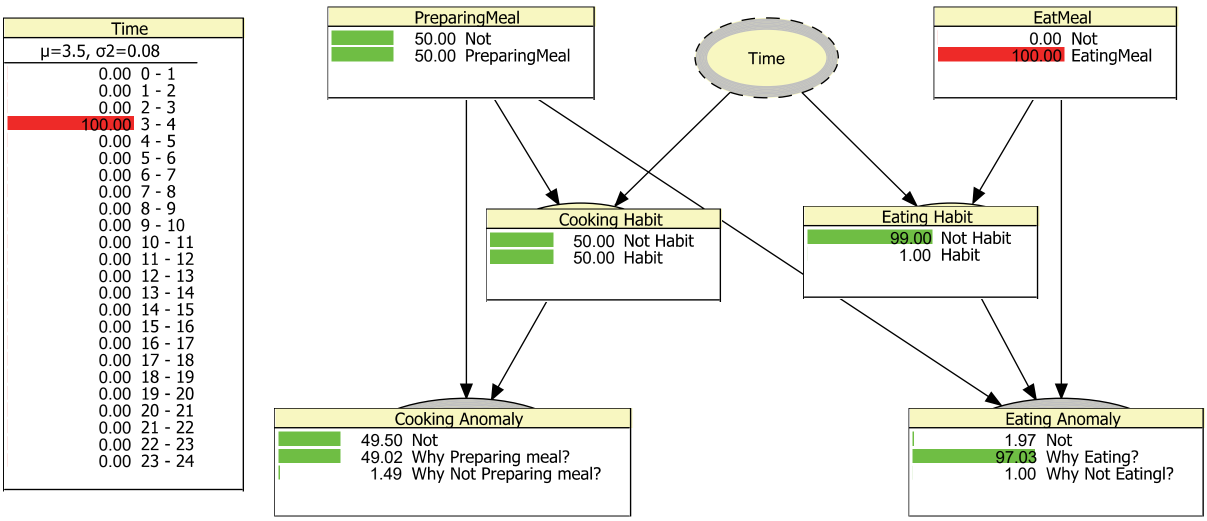


Figure 58, *The user starts eating during the night (time range between 3:00am and 4:00am).*

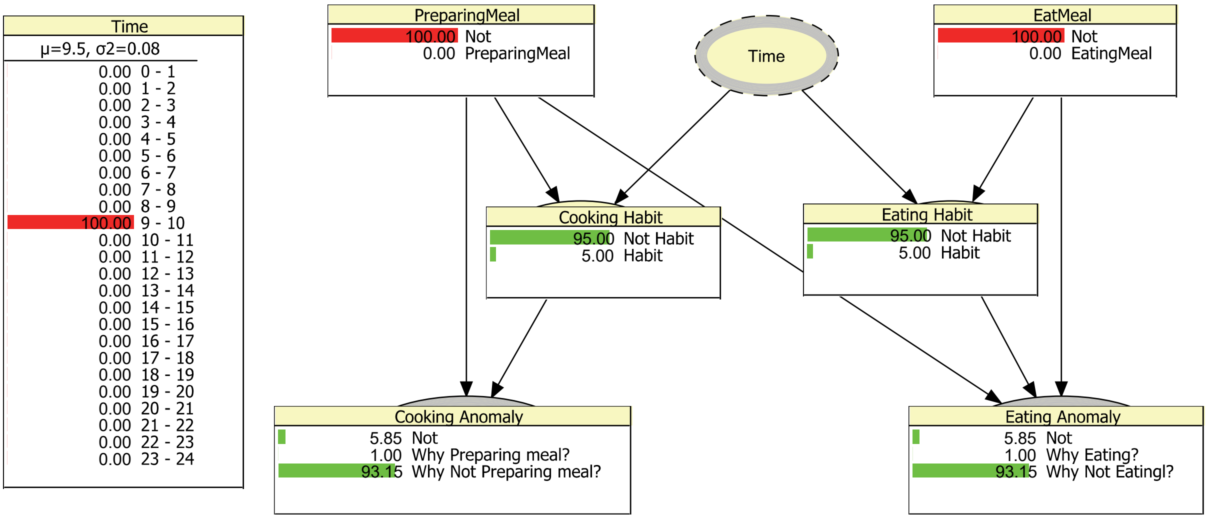


Figure 59, *The user has not made the breakfast yet, thus, he or she has not yet had breakfast (the time range between 9:00am and 10:00am).*

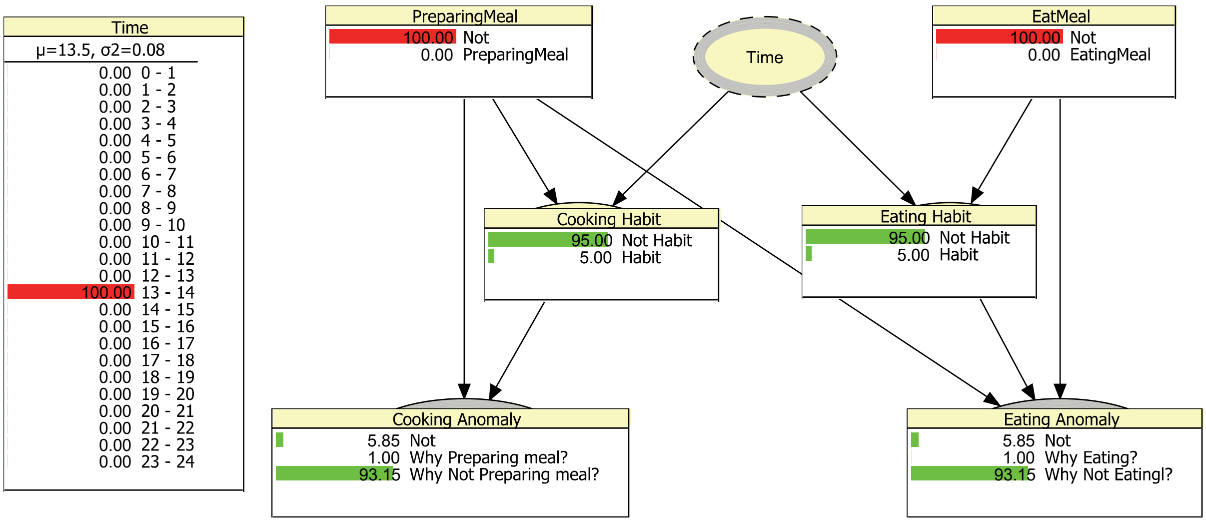


Figure 60, *The user has not prepared the lunch yet, thus, he or she has not yet had lunch (the time range between 1:00pm and 2:00pm).*

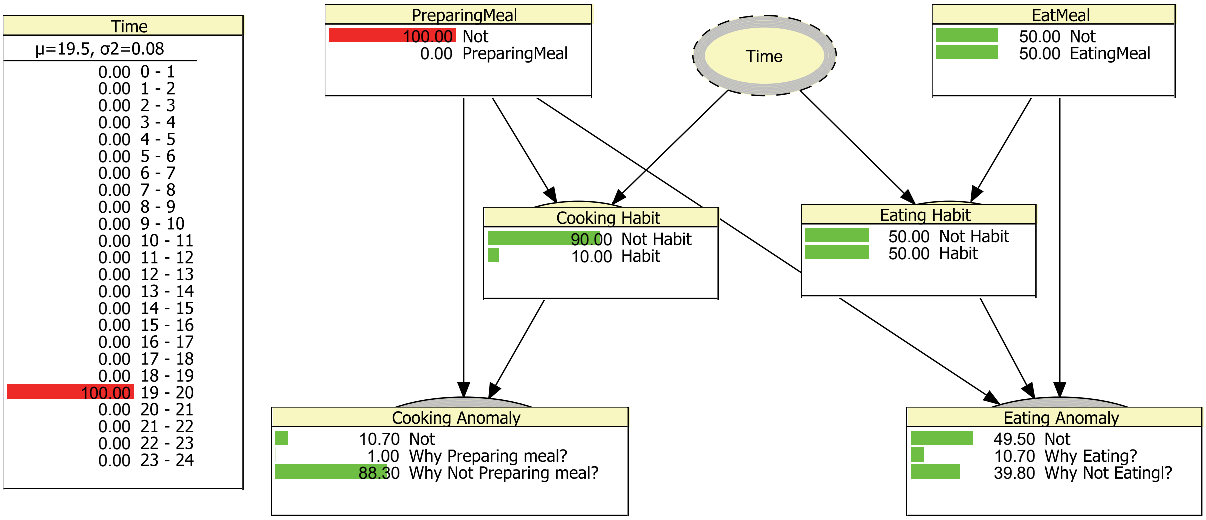


Figure 61, *The user has not prepared the dinner yet, thus, he or she has not yet had dinner (the time range between 7:00pm and 8:00pm).*

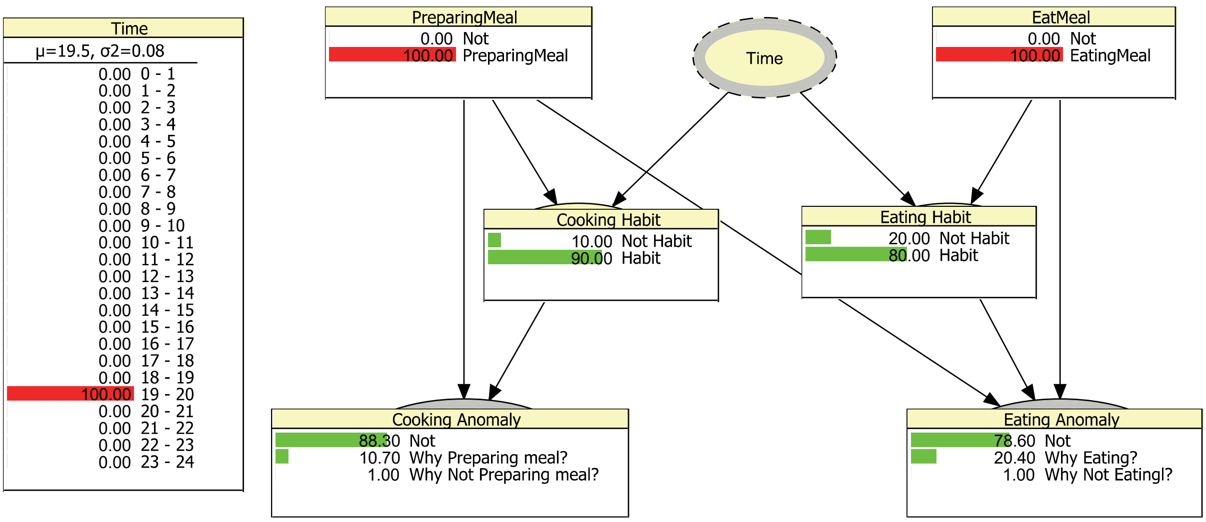


Figure 62, *The user has prepared the dinner and has had dinner (the time range between 7:00pm and 8:00pm). No anomaly in the behavioural pattern is detected indeed.*

5.2.3 Environmental comfort module

Some evidence has been manually provided to demonstrate the consistency of the decisions taken by this module in different scenarios *(Fig. 63 to 68)*.

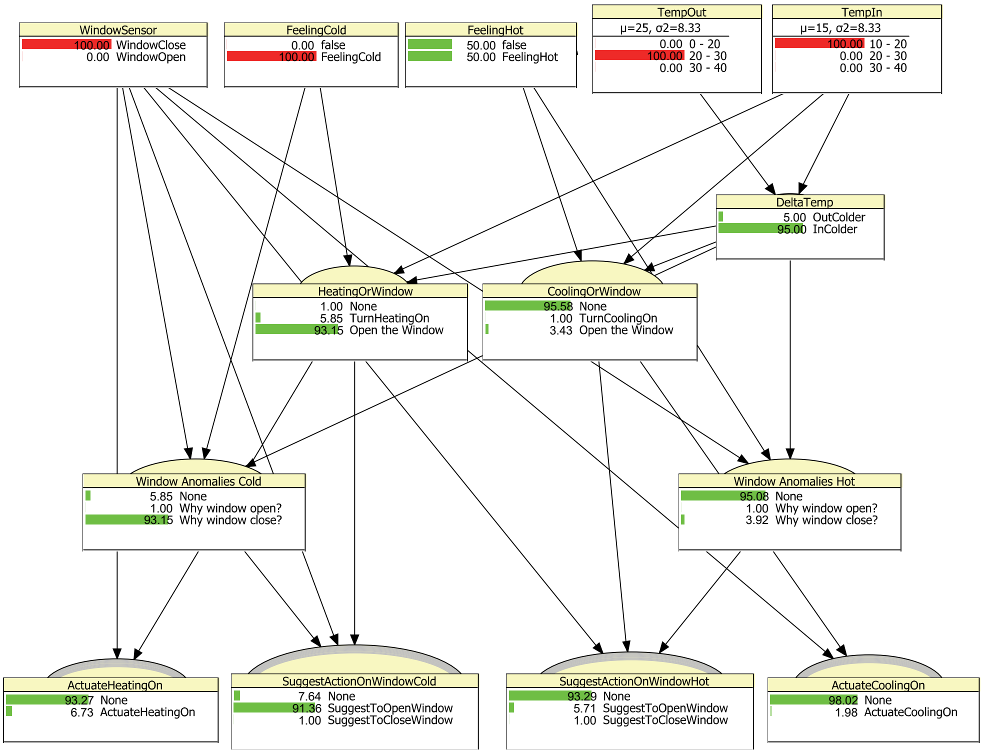


Figure 63, *The indoor temperature is lower than the outdoor one. The user is feeling cold, and the window is closed. The system suggests opening the window.*

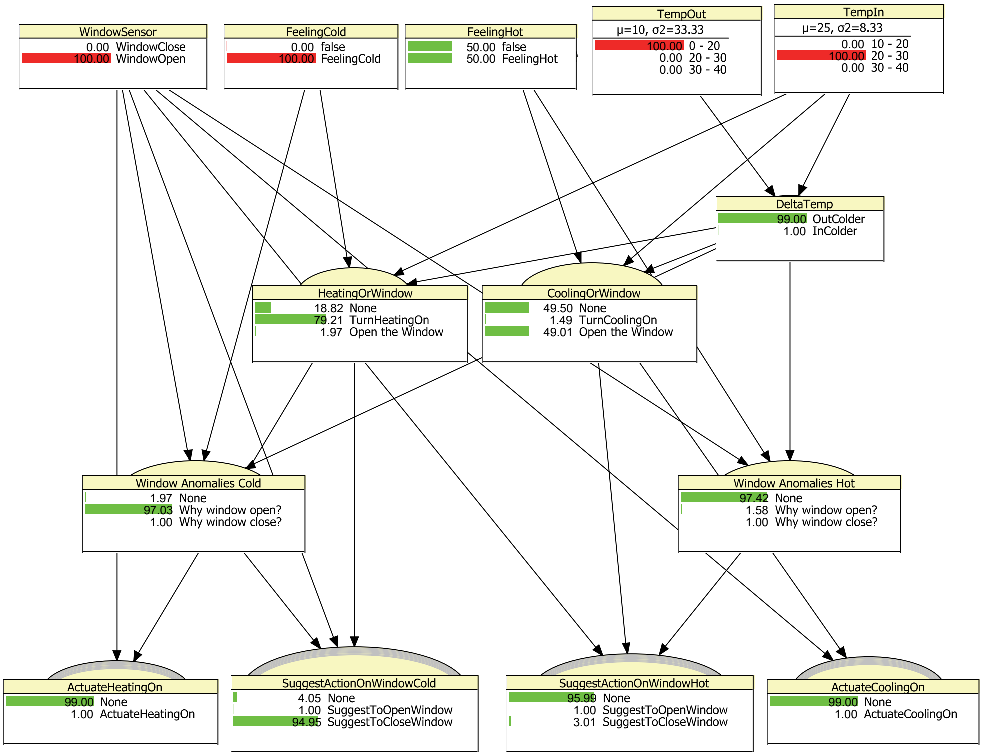


Figure 64, *The indoor temperature is rather higher than the outdoor one. The user is feeling cold, and the window is open. The system suggests closing the window.*

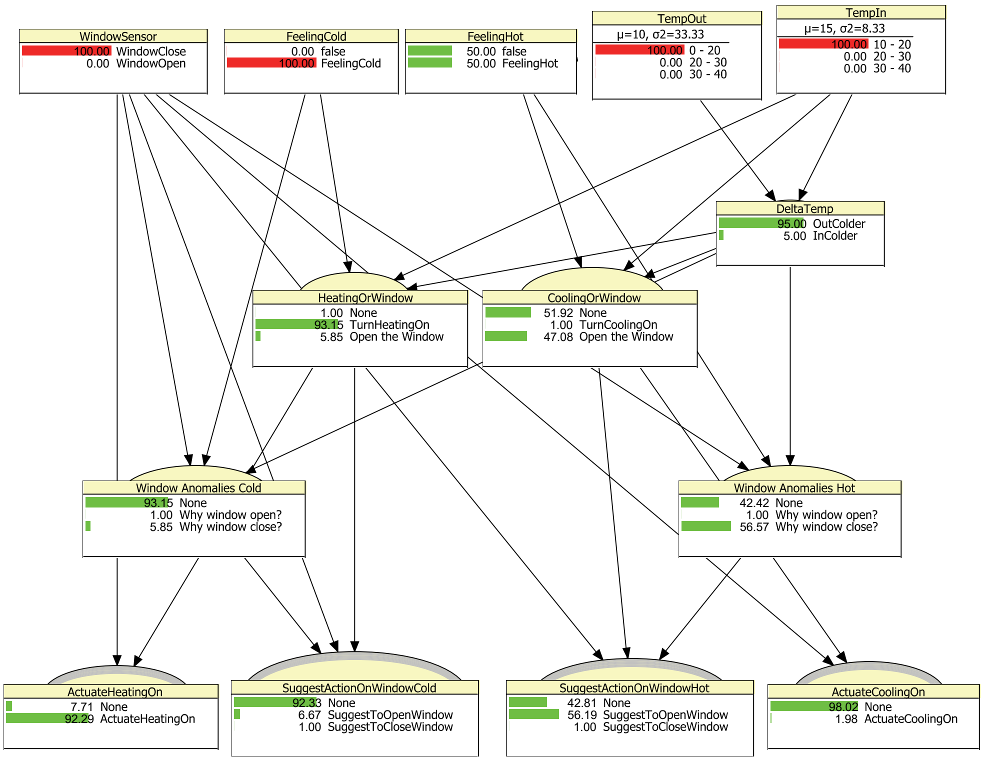


Figure 65, *The outdoor temperature is lower than the indoor one. The user is feeling cold, and the window has just been closed. The module actuates the heating system.*

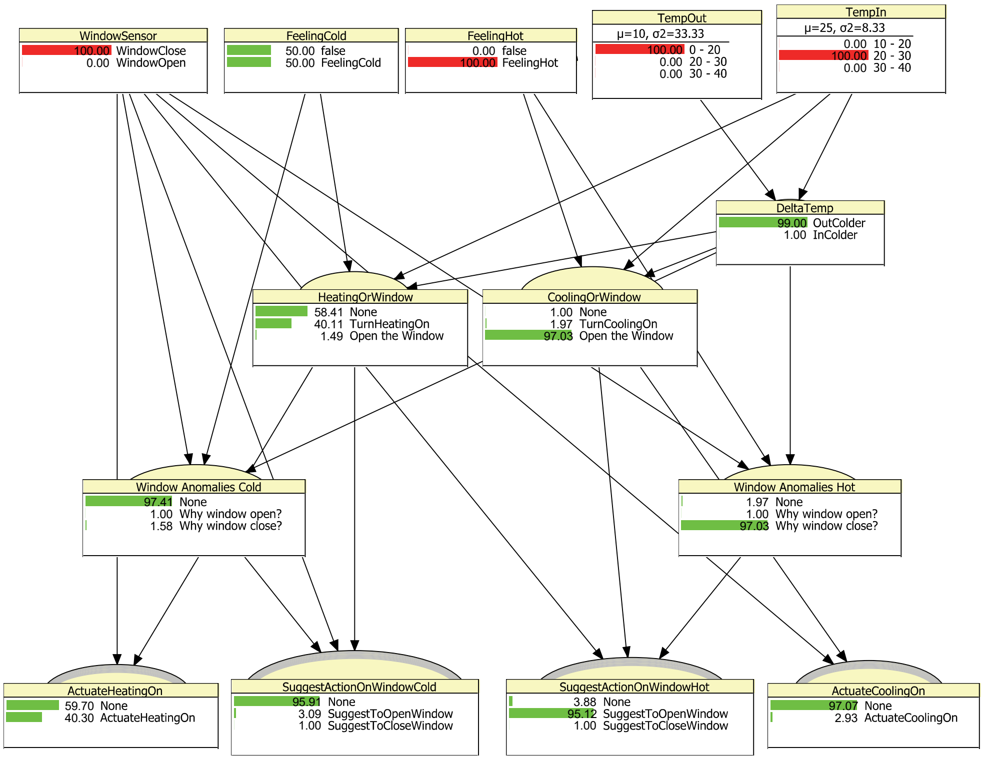


Figure 66, *The outdoor temperature is lower than the indoor one. The user is feeling hot, and the window is closed. The module suggests opening the window.*

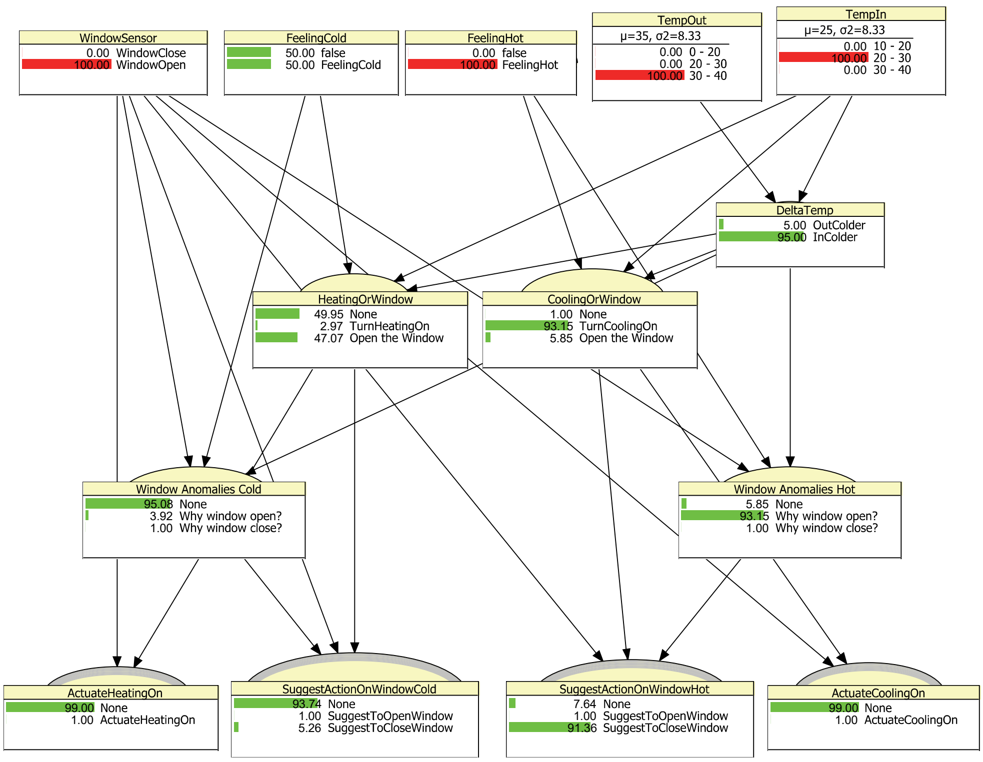


Figure 67, *The outdoor temperature is rather higher than the indoor one. The user is feeling hot, and the window is open. The module suggests closing the window.*

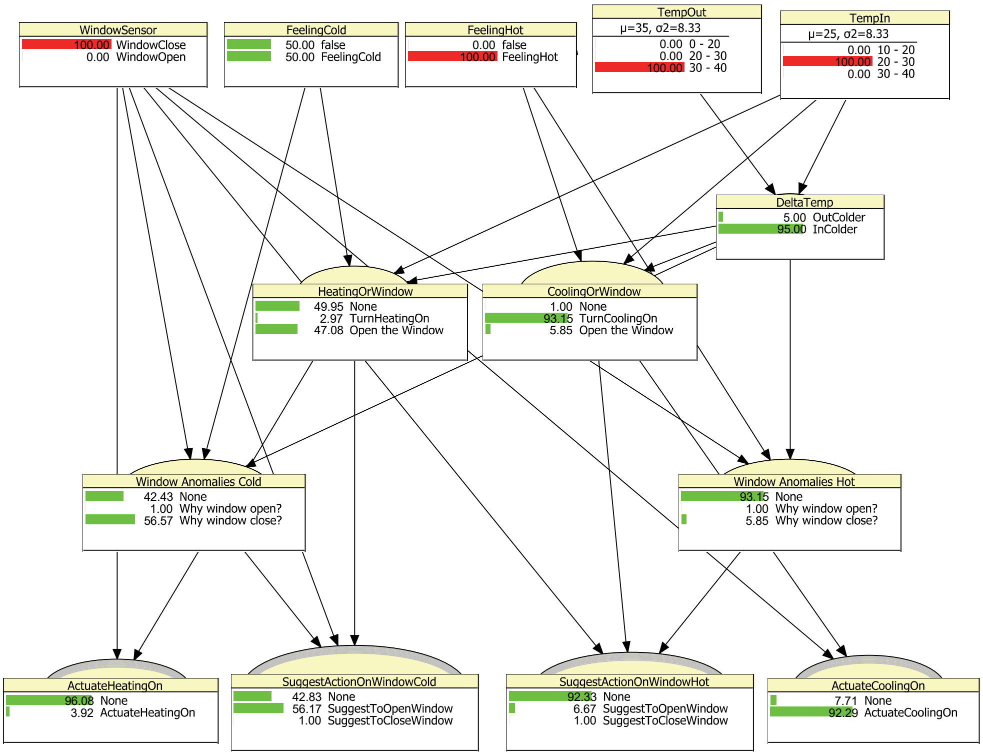


Figure 68, *The outdoor temperature is higher than the indoor one. The user is feeling hot, and the window has just been closed. The module actuates the cooling system.*

5.2.4 Emergency module

Some evidence has been manually provided to demonstrate the consistency of the decisions taken by this module in different scenarios *(Fig. 69)*.

Figure 69, *The user fell staggering and has chest pain. Direct emergency call triggered as well as symptom report.*

5.3 Bidirectional User-System Interaction

In this chapter are shown the processes the Dialog System can perform *(Fig. 70) (Fig. 71)*. The integration with the OOBN modules is omitted and left to future developments. It is worth noting that the system is capable of recognizing speech and reproducing audio in a significant number of different languages.

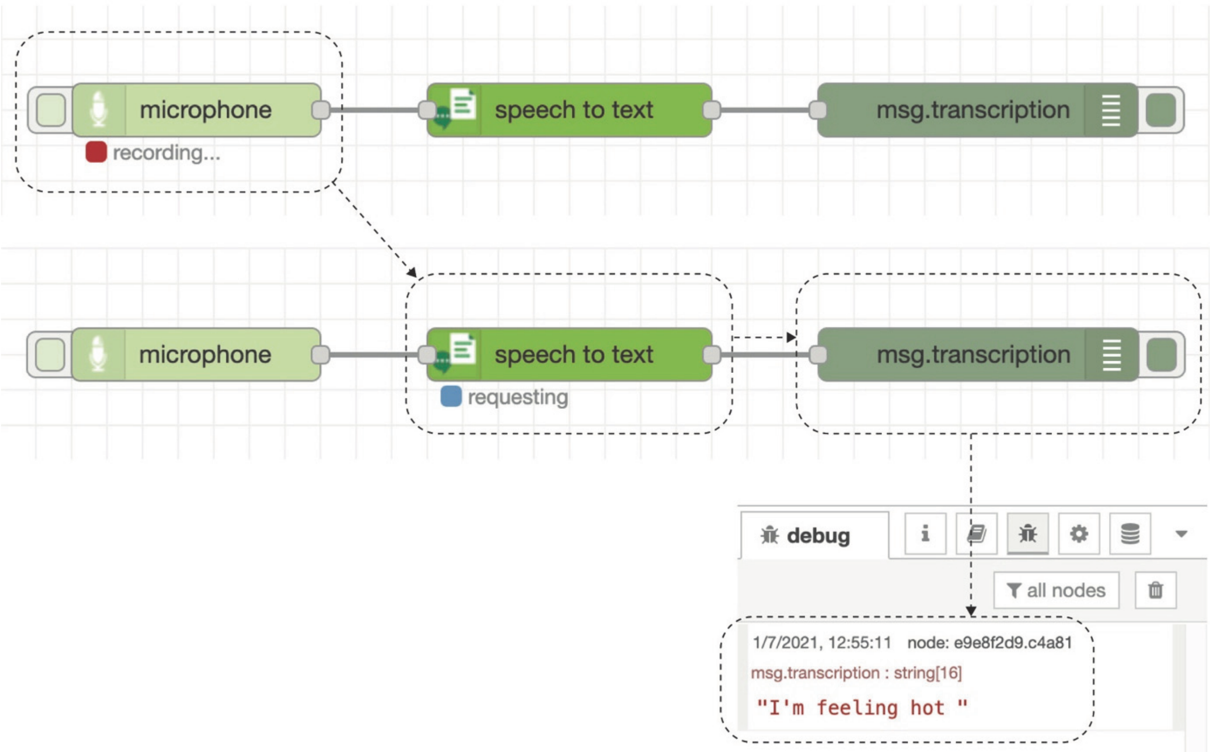


Figure 70, *Speech-To-Text process. First, the microphone node records the user's speech without requiring any Hot Phrase. Then, the speech is converted through the STT node that implicitly sends a request to the IBM's Watson STT service. Finally, a transcription of the speech is shown in the debug tab of the Node-Red editor. The transcript can also be directly generated as JSON to feed the OOBNs modules. The OOBNs can now exploit information from interactions to reason about the current scenario and make decisions.*

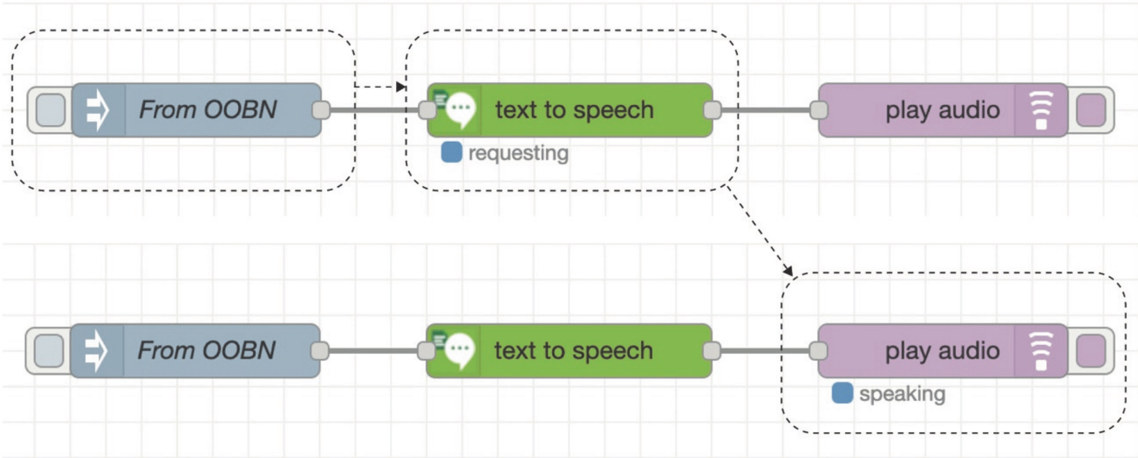


Figure 71, *Text-To-Speech process. This process could be triggered in two different scenarios: (a) the user has said something, the Reasoner make a decision depending on both the evidence and the speech, and then reply to the user; (b) the Reasoner make a decision depending on the evidence only, and then reply to the user. In both cases, the decision made by the OOBN modules are the input of the Dialog System's flow. Hence, the Reasoner is responsible for the tailored messages that should be played. These written messages are converted through the TTS node that implicitly sends a request to the IBM's Watson TTS service. Finally, the play audio node allows the converted speech to be played by the speaker.*

CHAPTER VI

CONCLUSION AND FUTURE DEVELOPMENTS

The focus of this dissertation lies in developing a prototype of a *Cognitive Building* that supports Ambient Assisted Living. Since the number of the elderly and, consequently, geriatric diseases such as cognitive disorders are increasing, a building-level Digital Twin in AAL scenarios offers solutions to support people in their older age. The end goal of our system is improving the quality of life of the elderly and disabled people, by preserving their autonomy whilst ensuring their safety. Our prototype is suitable for *Smart Environments*, such as new residential buildings, nursing homes, refurbished or retrofitted existing dwellings. The proposed Digital Twin is built upon five tiers, namely Data Acquisition Layer, Transmission Layer, Digital Modelling Layer, Data/Model Integration Layer and Service Layer. This grounded architecture defines a system able to autonomously perform high-level reasoning, that allows the detection of anomalies in daily scenarios and consequently offer support to the user. Anomalies can be caused by not correctly performed activities, their wrong combinations, temporal confusion, dangerous behaviours, and so forth. The strength of this system architecture lies in its development of knowledge: from raw data captured by multi-modal sensors (visual and non-visual) and their real-time 3D representation into the virtual counterpart, to high-level reasoning and decision-making once anomalies are detected. At the lower level, activity recognition is performed through Neural Networks that exploits 3D data deriving from the pre-processed real-time 3D representation of the user. At the higher level, Object-Oriented Bayesian Networks can recognize wasteful, nonsense or dangerous behaviours, environmental discomfort, changes in behavioural patterns, and serious medical situations or events, and thus trigger specific services. Any kind of interaction with the user, actuators and emergency calls are carried out through the Automation Hub, based on a VPL platform that acts as a broker of messages. The system accessibility and flexibility enable integrated tools and modules to be extended in future stages by adding further features if required.

We also consider a number of improvements, which can be addressed as the future work of this study:

- First, the use of larger 3D Skeleton dataset to train the Neural Network model. Currently, these datasets are still under development and allows the detection of a limited number of activities. Indeed, the Neural Network model exploited in our system is pre-trained on the NTU RGB+D 60 dataset, that includes only sixty user activities. Increasing their number, the reasoner would be able to detect more anomalies than those shown in this work.
- Another possible future work consists in developing the algorithm that allows data transfer and filtering between the Neural Network model and the Object-Oriented Bayesian Network. A filter should also be applied to sensor readings feeding the Bayesian Network.
- An improvement of the system should consider training the Bayesian Networks on data captured from a real AAL environment and working on a Bayesian Network Application Programming Interface (API) instead of a Graphical User Interface (GUI), that is the graphical and easy-to-visualize method exploited in this research. The API approach allows integration between Bayesian Networks and other tools.
- The connection between Bayesian Networks and the Automation Hub should be implemented.
- An archive of Object-Oriented Bayesian Networks concerning scenario awareness and anomalies recognition in AAL environments could be established. It will make available reusable and flexible modules that can be easily integrated in similar systems that aim at defining cognitive buildings. Sharing such a knowledge with the community will definitely foster development in this domain.
- The Automation Hub features could be extended through the integration of additional platforms of services and smart devices. For instance, home automation platforms such as Home Assistant could be linked to Node-Red.
- The whole pipeline should be fully tested and validated in an end-to-end manner.

This page intentionally left blank

APPENDIX A

CODES

Algorithm 1, Algorithm 2, Algorithm 3, Algorithm 4:

```
using UnityEngine;
using System;
using System.Collections.Generic;
using System.IO;
using UnityEditor;
using System.Globalization;

/*using System.Linq;
public static class ExtensionMethods
{
    public static Transform[] FindChildren(this Transform transform, string name)
    {
        return transform.GetComponentsInChildren<Transform>().Where(t => t.name ==
name).ToArray();
    }
}*/

public static class MyExtensions
{
    /// <summary>
    /// Returns the left part of this string instance.
    /// </summary>
    /// <param name="count">Number of characters to return.</param>
    public static string Left(this string input, int count)
    {
        return input.Substring(0, Math.Min(input.Length, count));
    }
    public static string Right(this string input, int count)
    {
        return input.Substring(Math.Max(input.Length - count, 0), Math.Min(count,
input.Length));
    }
}

public class ManAvatar : MonoBehaviour
{
    [Header("Log file for joints data")]
    public string logFile = "Assets/Resources/joints_log.txt";
    public GameObject targetGameObject;
    readonly Dictionary<int, string> exportList = new Dictionary<int, string>() // Dictionary used
to map NTU_RGB+D Skeleton dataset with avatar joints
    {
        {1 , "Root_M"},
        {2 , "Spine1_M"},
        {3 , "Neck_M"},
        {4 , "Head_M"},
        {5 , "Shoulder_L"},
```

```
        {6 , "Elbow_L"},
        {7 , "Wrist_L"},
        {8 , "MiddleFinger1_L"},
        {9 , "Shoulder_R"},
        {10 , "Elbow_R"},
        {11 , "Wrist_R"},
        {12 , "MiddleFinger1_R"},
        {13 , "Hip_L"},
        {14 , "Knee_L"},
        {15 , "Ankle_L"},
        {16 , "Toes_L"},
        {17 , "Hip_R"},
        {18 , "Knee_R"},
        {19 , "Ankle_R"},
        {20 , "Toes_R"},
        {21 , "Scapula_M"}, // This does not exist but is computed as mean between Scapula_L
& Scapula_R
        {22 , "MiddleFinger4_L"},
        {23 , "ThumbFinger4_L"},
        {24 , "MiddleFinger4_R"},
        {25 , "ThumbFinger4_R"},
    };

    [Header("Avatar joints mapping")]
    [SerializeField] ModelJoint[] modelJoints;
    [SerializeField] nuitrack.JointType rootJoint = nuitrack.JointType.Torso;

    /// <summary> Model bones </summary>
    readonly Dictionary<nuitrack.JointType, ModelJoint> jointsMapping = new
Dictionary<nuitrack.JointType, ModelJoint>();

    // Estimated height of skeleton
    float height=1.8f; // m

    void Start()
    {
        // Initialize Key-Value structure that maps nuitrack joint types with modelJoints
        for (int i = 0; i < modelJoints.Length; i++)
        {
            modelJoints[i].baseRotOffset = modelJoints[i].bone.rotation; // Setup base Rot Offset
            jointsMapping.Add(modelJoints[i].jointType.TryGetMirrored(), modelJoints[i]);
        }
    }

    void Update()
    {
        if (CurrentUserTracker.CurrentSkeleton != null)
ProcessSkeleton(CurrentUserTracker.CurrentSkeleton);
    }

    void ProcessSkeleton(nuitrack.Skeleton skeleton)
    {
        float thr = 0.1f; // Detection threshold
        float a = 0.02f; // Weight of innovation for filtering height
        float t = 0.2f; // Weight of innovation for filtering position and rotation of bones
```

```
        // Scale avatar size based on skeleton data
        // avatar height is 1.8m and it must be scaled based on sum of skeleton bones
        float newHeight = GetHeight(skeleton);
        if (!float.IsNaN(newHeight))
        {
            // filter height value
            height = (1-a)*height + a*newHeight;
            //Debug.Log("---> Height=" + height);
            GameObject.Find("TextBox1").GetComponent<UnityEngine.UI.Text>().text =
"Estimated Height = " + height.ToString("0.00") + " m";
            // scale avatar to match human height
            targetGameObject.transform.localScale = new Vector3(height/1.80f, height/1.80f,
height/1.80f); // scale with respect avatar height that is 1.8m
        }

        //Calculate the model position: take the root position and invert movement along the Z
axis
        Vector3 rootPos = Quaternion.Euler(0f, 180f, 0f) * (0.001f *
skeleton.GetJoint(rootJoint).ToVector3());
        if (skeleton.GetJoint(rootJoint).Confidence >= thr)
            transform.position = (1-t) * transform.position + t * rootPos;

        // Scroll all avatar joints and move avatar based on skeleton detected by nuitrack
        foreach (var joint in jointsMapping)
        {
            //Get corresponding joint from the Nuitrack
            nuitrack.Joint jointNuitrack = skeleton.GetJoint(joint.Key);

            // Check if confidence is grater than threshold
            if (jointNuitrack.Confidence >= thr)
            {
                // Get joint of the avatar corresponding to selected nuitrack joint
                ModelJoint avatarJoint = joint.Value;
                //Calculate the model bone rotation: take the mirrored joint orientation, add a basic
rotation of the model bone, invert movement along the Z axis
                Quaternion jointOrient = Quaternion.Inverse(CalibrationInfo.SensorOrientation) *
(jointNuitrack.ToQuaternionMirrored()) * avatarJoint.baseRotOffset;
                // Apply rotation to bone based on the detected skeleton
                avatarJoint.bone.rotation = Quaternion.Slerp(avatarJoint.bone.rotation, jointOrient,
t);
            }
        }

        // log skeleton data in text file
        StreamWriter writer = new StreamWriter(logFile, true);
        // NTU-RGB+D Prehamble (space delimited file)
        // N° of bodies
        writer.WriteLine("1"); // TODO: set to 0 when no skeleton found!!!
        // bodyID clipedEdges handLeftConfidence handLeftState   handRightConfidence
handRightState  isResticted leanX   leanY trackingState
        writer.WriteLine("1111111111111111 0 1 1 1 1 0 0.0 0.0 2");
        // jointCount
        writer.WriteLine("25"); // number of joints
        // joints
        foreach (var item in exportList)
        {
```

```
            Vector3 pos = new Vector3(float.NaN, float.NaN, float.NaN);
            try
            {
                GameObject go = GameObject.Find(item.Value).gameObject;
                pos = go.transform.position;
            }
            catch (Exception)
            {
                // when middle joint do not exist, computes it as the mean point between Left and
Right joints
                if (item.Value.Right(1) == "M")
                {
                    // Look for position of left joint with same name
                    pos = GameObject.Find(item.Value.Left(item.Value.Length - 1) +
"L").gameObject.transform.position;
                    // Look for position of right joint with same name and sum
                    pos += GameObject.Find(item.Value.Left(item.Value.Length - 1) +
"R").gameObject.transform.position;
                    // Compute mean point
                    pos /= 2;
                }
            }
            if (pos.x != float.NaN)
            {
                // Set decimal separator to "." to be used when converting numbers to strings
                NumberFormatInfo nfi = new NumberFormatInfo();
                nfi.NumberDecimalSeparator = ".";
                // TODO: Convert Unity to NTU coordinates system if needed (probably not
needed)
                // ...
                // x y z depthXdepthY   colorX colorY orientationW  orientationXorientationY
  orientationZ trackingState
                writer.WriteLine(pos.x.ToString(nfi) + " " + pos.y.ToString(nfi) + " " +
pos.z.ToString(nfi) + " 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 2");
            }
            else
            {
                // Missing joint
                writer.WriteLine("0 0 0 NaN NaN NaN NaN 0 1 0 0 0");
            }
            //Debug.Log(" JOINT #" + item.Key + " | " + item.Value + " | Position=" + pos);
        }
        writer.Close();
    }

    private float GetHeight(nuitrack.Skeleton skeleton)
    {
        /// Returns skeleton height in meters
        /// |Knee_L- Ankle_L | + |Hip_L-Knee_L| + | Spine1_M-Root_M | + | Scapula_M-
Spine1_M | + |neck-collar| + |head-neck|
        float thr = 0.2f; // confidenze below which it returns NaN height
        if (skeleton.GetJoint(nuitrack.JointType.Head).Confidence >= thr &
            skeleton.GetJoint(nuitrack.JointType.Neck).Confidence >= thr &
            skeleton.GetJoint(nuitrack.JointType.LeftCollar).Confidence >= thr &
            skeleton.GetJoint(nuitrack.JointType.Torso).Confidence >= thr &
            skeleton.GetJoint(nuitrack.JointType.Waist).Confidence >= thr &
```

```
            skeleton.GetJoint(nuitrack.JointType.LeftHip).Confidence >= thr &
            skeleton.GetJoint(nuitrack.JointType.LeftKnee).Confidence >= thr &
            skeleton.GetJoint(nuitrack.JointType.LeftAnkle).Confidence >= thr)
        {
            float height = 0;
            height +=
Vector3.Magnitude(skeleton.GetJoint(nuitrack.JointType.Head).ToVector3() -
skeleton.GetJoint(nuitrack.JointType.Neck).ToVector3()) * 2.4f; // for considering remaining
part of head
            height += Vector3.Magnitude(skeleton.GetJoint(nuitrack.JointType.Neck).ToVector3()
- skeleton.GetJoint(nuitrack.JointType.LeftCollar).ToVector3());
            height +=
Vector3.Magnitude(skeleton.GetJoint(nuitrack.JointType.LeftCollar).ToVector3() -
skeleton.GetJoint(nuitrack.JointType.Torso).ToVector3());
            height +=
Vector3.Magnitude(skeleton.GetJoint(nuitrack.JointType.Torso).ToVector3() -
skeleton.GetJoint(nuitrack.JointType.Waist).ToVector3());
            height +=
Vector3.Magnitude(skeleton.GetJoint(nuitrack.JointType.Waist).ToVector3() -
skeleton.GetJoint(nuitrack.JointType.LeftHip).ToVector3());
            height +=
Vector3.Magnitude(skeleton.GetJoint(nuitrack.JointType.LeftHip).ToVector3() -
skeleton.GetJoint(nuitrack.JointType.LeftKnee).ToVector3());
            height +=
Vector3.Magnitude(skeleton.GetJoint(nuitrack.JointType.LeftKnee).ToVector3() -
skeleton.GetJoint(nuitrack.JointType.LeftAnkle).ToVector3());
            return height / 1000;
        }
        else
            return float.NaN;
    }
}
```

Algorithm 5:

```
function []=show_skeleton_3d(skeletonfilename)
% Draws the skeleton data in 3D.
%
% Usage:
%   show_skeleton_3d(skeletonfilename)
%
% Example:
%   show_skeleton_3d('S001C001P001R001A001.skeleton');
%
% Coded by Massimo Vaccarini
% UNIVPM
% 19/05/2021

bodyinfo = read_skeleton_file('joints_log_1.skeleton');

% in the skeleton structure, each joint is connected to some other joint:
connecting_joint = ...
  [2 1 21 3 21 5 6 7 21 9 10 11 1 13 14 15 1 17 18 19 2 8 8 12 12];
```

```
figure;
axis equal;
hold on;
view([45 30]);
x1=zeros(25,1);
y1=zeros(25,1);
z1=zeros(25,1);
x2=zeros(25,1);
y2=zeros(25,1);
z2=zeros(25,1);

% reapeat this for every frame
for f=1:numel(bodyinfo)
    try
        % for all the detected skeletons in the current frame:
        for b=1:numel(bodyinfo(f).bodies)
            % for all the 25 joints within each skeleton:
            for j=1:25
                try
                    k = connecting_joint(j);
                    joint = bodyinfo(f).bodies(b).joints(j);
                    x1(j) = joint.x;
                    y1(j) = joint.z;
                    z1(j) = joint.y;
                    joint2 = bodyinfo(f).bodies(b).joints(k);
                    x2(j) = joint2.x;
                    y2(j) = joint2.z;
                    z2(j) = joint2.y;
                catch err1
                    disp(err1);
                end
            end
        end
        cla;
        plot3([x1, x2]',[y1, y2]',[z1, z2]','ro-');
        pause(0.01);
    catch err2
        disp(err2);
    end
end
end
```

Algorithm 6: developed in (Liu, Zhang, Chen, Wang, & Ouyang, 2020), available at: https://github.com/kenziyuliu/MS-G3D/blob/master/model/msg3d.py

```
import sys
sys.path.insert(0, '')
import math
import numpy as np
import torch
import torch.nn as nn
import torch.nn.functional as F

from utils import import_class, count_params
from model.ms_gcn import MultiScale_GraphConv as MS_GCN
```

```
from model.ms_tcn import MultiScale_TemporalConv as MS_TCN
from model.ms_gtcn import SpatialTemporal_MS_GCN, UnfoldTemporalWindows
from model.mlp import MLP
from model.activation import activation_factory
class MS_G3D(nn.Module):
    def __init__(self,
                 in_channels,
                 out_channels,
                 A_binary,
                 num_scales,
                 window_size,
                 window_stride,
                 window_dilation,
                 embed_factor=1,
                 activation='relu'):
        super().__init__()
        self.window_size = window_size
        self.out_channels = out_channels
        self.embed_channels_in = self.embed_channels_out = out_channels // embed_factor
        if embed_factor == 1:
            self.in1x1 = nn.Identity()
            self.embed_channels_in = self.embed_channels_out = in_channels
            # The first STGC block changes channels right away; others change at collapse
            if in_channels == 3:
                self.embed_channels_out = out_channels
        else:
            self.in1x1 = MLP(in_channels, [self.embed_channels_in])
        self.gcn3d = nn.Sequential(
            UnfoldTemporalWindows(window_size, window_stride, window_dilation),
            SpatialTemporal_MS_GCN(
                in_channels=self.embed_channels_in,
                out_channels=self.embed_channels_out,
                A_binary=A_binary,
                num_scales=num_scales,
                window_size=window_size,
                use_Ares=True
            )
        )
        self.out_conv = nn.Conv3d(self.embed_channels_out, out_channels, kernel_size=(1,
self.window_size, 1))
        self.out_bn = nn.BatchNorm2d(out_channels)
    def forward(self, x):
        N, _, T, V = x.shape
        x = self.in1x1(x)
        # Construct temporal windows and apply MS-GCN
        x = self.gcn3d(x)
        # Collapse the window dimension
        x = x.view(N, self.embed_channels_out, -1, self.window_size, V)
        x = self.out_conv(x).squeeze(dim=3)
        x = self.out_bn(x)
        # no activation
        return x
class MultiWindow_MS_G3D(nn.Module):
    def __init__(self,
                 in_channels,
                 out_channels,
```

```
                A_binary,
                num_scales,
                window_sizes=[3,5],
                window_stride=1,
                window_dilations=[1,1]):

        super().__init__()
        self.gcn3d = nn.ModuleList([
            MS_G3D(
                in_channels,
                out_channels,
                A_binary,
                num_scales,
                window_size,
                window_stride,
                window_dilation
            )
            for window_size, window_dilation in zip(window_sizes, window_dilations)
        ])
    def forward(self, x):
        # Input shape: (N, C, T, V)
        out_sum = 0
        for gcn3d in self.gcn3d:
            out_sum += gcn3d(x)
        # no activation
        return out_sum
class Model(nn.Module):
    def __init__(self,
                num_class,
                num_point,
                num_person,
                num_gcn_scales,
                num_g3d_scales,
                graph,
                in_channels=3):
        super(Model, self).__init__()
        Graph = import_class(graph)
        A_binary = Graph().A_binary
        self.data_bn = nn.BatchNorm1d(num_person * in_channels * num_point)
        # channels
        c1 = 96
        c2 = c1 * 2     # 192
        c3 = c2 * 2     # 384
        # r=3 STGC blocks
        self.gcn3d1 = MultiWindow_MS_G3D(3, c1, A_binary, num_g3d_scales,
window_stride=1)
        self.sgcn1 = nn.Sequential(
            MS_GCN(num_gcn_scales, 3, c1, A_binary, disentangled_agg=True),
            MS_TCN(c1, c1),
            MS_TCN(c1, c1))
        self.sgcn1[-1].act = nn.Identity()
        self.tcn1 = MS_TCN(c1, c1)
        self.gcn3d2 = MultiWindow_MS_G3D(c1, c2, A_binary, num_g3d_scales,
window_stride=2)
        self.sgcn2 = nn.Sequential(
            MS_GCN(num_gcn_scales, c1, c1, A_binary, disentangled_agg=True),
```

```
            MS_TCN(c1, c2, stride=2),
            MS_TCN(c2, c2))
        self.sgcn2[-1].act = nn.Identity()
        self.tcn2 = MS_TCN(c2, c2)
        self.gcn3d3 = MultiWindow_MS_G3D(c2, c3, A_binary, num_g3d_scales,
window_stride=2)
        self.sgcn3 = nn.Sequential(
            MS_GCN(num_gcn_scales, c2, c2, A_binary, disentangled_agg=True),
            MS_TCN(c2, c3, stride=2),
            MS_TCN(c3, c3))
        self.sgcn3[-1].act = nn.Identity()
        self.tcn3 = MS_TCN(c3, c3)
        self.fc = nn.Linear(c3, num_class)
    def forward(self, x):
        N, C, T, V, M = x.size()
        x = x.permute(0, 4, 3, 1, 2).contiguous().view(N, M * V * C, T)
        x = self.data_bn(x)
        x = x.view(N * M, V, C, T).permute(0,2,3,1).contiguous()

        # Apply activation to the sum of the pathways
        x = F.relu(self.sgcn1(x) + self.gcn3d1(x), inplace=True)
        x = self.tcn1(x)
        x = F.relu(self.sgcn2(x) + self.gcn3d2(x), inplace=True)
        x = self.tcn2(x)
        x = F.relu(self.sgcn3(x) + self.gcn3d3(x), inplace=True)
        x = self.tcn3(x)
        out = x
        out_channels = out.size(1)
        out = out.view(N, M, out_channels, -1)
        out = out.mean(3)   # Global Average Pooling (Spatial+Temporal)
        out = out.mean(1)   # Average pool number of bodies in the sequence
        out = self.fc(out)
        return out
if __name__ == "__main__":
    # For debugging purposes
    import sys
    sys.path.append('..')
    model = Model(
        num_class=60,
        num_point=25,
        num_person=2,
        num_gcn_scales=13,
        num_g3d_scales=6,
        graph='graph.ntu_rgb_d.AdjMatrixGraph'
    )
    N, C, T, V, M = 6, 3, 50, 25, 2
    x = torch.randn(N,C,T,V,M)
    model.forward(x)
    print('Model total # params:', count_params(model))
```

This page intentionally left blank

REFERENCES

AAL Association. (2008). *AAL Programme*. Retrieved from AAL Programme: http://www.aal-europe.eu/

Barb, A., & Shyu, C.-R. (2008). *Knowledge Representation and Exchange of Visual Patterns using Semantic Abstractions.* University of Missouri, Computer Science Department.

Berryhill, M., Peterson, D., Jones, K., & Tanoue, R. (2012). Cognitive Disorders. *Encyclopedia of Human Behavior: Second Edition*, 536-542.

Bolton, A., Enzer, M., Schooling, J., & al., e. (2018). *The Gemini Principles: Guiding values for the national digital twin and information management framework.* Centre for Digital Built Britain and Digital Framework Task Group.

buildingSMART International. (2020). *Enabling an Ecosystem of Digital Twins.*

Calderita, L., Vega, A., Barroso-Ramírez, S., Bustos, P., & Núñez, P. (2020). Designing a cyber-physical system for ambient assisted living: A use-case analysis for social robot navigation in caregiving centers. *Sensors (Switzerland), 20*(14), 1-25.

Chevallier, Z., Finance, B., & Cohen Boulakia, B. (2020). A Reference Architecture for Smart Building Digital Twin. *SeDiT@ESWC*.

Das, S., Sharma, S., Dai, R., Bremond, F., & Thonnat, M. (2020). *VPN: Learning Video-Pose Embedding for Activities of Daily Living.* Retrieved from arXiv: arXiv:2007.03056v1

De Grassi, M., Naticchia, B., Giretti, A., & Carbonari, A. (2009). *Reti Bayesiane con applicazioni all'edilizia e alla gestione del territorio.* Milano: Franco Angeli.

De Paola, A., Ferraro, P., Gaglio, S., Lo Re, G., & Das, S. (2017). An Adaptive Bayesian System for Context-Aware Data Fusion in Smart Environments. *IEEE Transaction on Mobile Computing, 16*(6), 1502-1515.

De Paola, A., Ferraro, P., Gaglio, S., Re, G., Morana, M., Ortolani, M., & Peri, D. (2017). An ambient intelligence system for assisted living. *AEIT International Annual Conference*, (pp. 1-6). Cagliari, Italy.

El Saddik, A. (2018). Digital Twins: The Convergence of Multimedia Technologies. *IEEE MultiMedia, 25*(02), 87-92.

Endsley, M. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors, 37*(1), 32-64.

Fox, J. (2017). Cognitive systems at the point of care: The CREDO program. *Journal of Biomedical Informatics, 68*, 83-95.

Gárate, A., Herrasti, N., & López, A. (2005). GENIO: An Ambient Intelligence application in home automation and entertainment environment. *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, (pp. 241–245). Grenoble, France.

Girshick, R., Donahue, J., Darrel, T., & Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation.* UC Berkeley.

Hippold, S. (2019, Jan 23). *How Digital Twins Simplify the IoT.* Retrieved from Gartner: https://www.gartner.com/smarterwithgartner/how-digital-twins-simplify-the-iot/

Hugin. (n.d.). Retrieved from Hugin: https://hugin.com/wp-content/uploads/2016/05/Building-a-BN-Tutorial.pdf

IBM Cloud Education. (2020, Aug 17). *Neural Networks*. Retrieved from IBM: https://www.ibm.com/cloud/learn/neural-networks

Jayamani, J., Thangaraju, P., Thangaraju, E., & Venkatesan, S. (2020). Decentralisation of healthcare system due to COVID-19 and its impact on hospital based laboratories - Pandemic panic patients' reflection? *Journal of Responsible Technology*, 1.

json.org. (n.d.). Retrieved from Introducing JSON: https://www.json.org/json-en.html

Ke, Q., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2017). A New Representation of Skeleton Sequences for 3D Action Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 4570-4579). Honolulu, HI, USA.

Khajavi, S., Motlagh, N., Jaribion, A., Werner, L., & Holmstrom, J. (2019). Digital Twin: Vision, benefits, boundaries, and creation for buildings. *IEEE Access, 7*.

Laraba, S., Tilmanne, J., & Dutoit, T. (2019). Leveraging Pre-trained CNN Models for Skeleton-Based Action Recognition. In D. Tzovaras, D. Giakoumis, M. Vincze, & A. Argyros, *Computer Vision Systems. ICVS 2019. Lecture Notes in Computer Science* (Vol. 11754). Springer, Cham.

Leonardi, C., Mennecozzi, C., Not, E., Pianesi, F., & Zancarano, M. (2008). Supporting older adults social network: the design of e-inclusion communication services. *Gerontechnology, 7*(2), 153-153.

Li, B., Chen, H., Yucheng, C., Yuchao, D., & Mingyi, H. (2017). Skeleton Boxes: Solving skeleton based action detection with a single Deep Convolutional Neural Network. *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, (pp. 613-616). Hong Kong, China.

Li, C., Zhong, Q., Xie, D., & Pu, S. (2017). Skeleton-based Action Recognition with Convolutional Neural Networks. *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, (pp. 597-600). Hong Kong, China.

Li, C., Zhong, Q., Xie, D., & Shiliang, P. (2018). *Co-occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation.* arXiv:1804.06055. Retrieved from arXiv:1804.06055

Li, R., Lu, B., & McDonald-Maier, K. (2015). Cognitive assisted living ambient system: a survey. *Digital Communications and Networks, 1*, 229-252.

Lim, K., Zheng, P., & Chen, C. (2020). A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives. *Journal of Intelligent Manufacturing*, 1313–1337.

Liu, Z., Zhang, H., Chen, Z., Wang, Z., & Ouyang, W. (2020). *Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition.* arXiv:2003.14111v2.

Lu, Q., A.M.ASCE, Parlikad, A., Woodall, P., Don Ranasinghe, G., Xie, X., . . . Schooling, J. (2020). Developing a Digital Twin at Building and City Levels: Case Study of West Cambridge Campus. *Journal of Management in Engineering, 36*(3).

Lu, Q., Xie, X., Parlikad, A., Schooling, J., & Konstantinou, E. (2020). Moving from Building Information Models to Digital Twins for Operation and Maintenance. *Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction*, 1-9.

Mao, D., Lin, X., Liu, Y., Xu, M., Wang, G., Chen, J., & Zhang, W. (2021). Activity Recognition from Skeleton and Acceleration Data Using CNN and GCN. In M. Ahad, P. Lago, & S. Inoue, *Human Activity Recognition Challenge. Smart Innovation, Systems and Technologies* (Vol. 199, pp. 15-25). Singapore: Springer.

MedPro Group. (2018, Jan 01). *Communication in the Diagnostic Process, How Breakdowns and Missed Opportunities Can Lead to Errors — and What You Can Do About Them.* (L. M. Cascella, Ed.) Retrieved from Attorney Protective: https://www.attorneyprotective.com/documents/10502/2820774/Communication+in+the+Diagnostic+Process.pdf

Nini, M. (2020, 01 29). *Situational awareness: What it is and why it matters as a management tool*. Retrieved from CQ Net: https://www.ckju.net/en/dossier/situational-awareness-what-it-and-why-it-matters-management-tool

Ordóñez, F., & Roggen, D. (2016). Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors, 16*(1), 115.

Patel, A., & Shah, J. (2020). Real-time human behaviour monitoring using hybrid ambient assisted living framework. *Journal of Reliable Intelligent Environments, 6*(2), 95-106.

Pazienza, A., Mallardi, G., Fasciano, C., & Vitulano, F. (2020). *Artificial Intelligence on Edge Computing: a Healthcare Scenario in Ambient Assisted Living.* Polytechnic University of Bari and Innovation Lab, Exprivia S.p.A.

Pearl, J., & Mackenzie, D. (2019). *The Book of Why: The New Science of Cause and Effect.* Penguin.

Pearl, J., Glymour, M., & Jewell, N. (2016). *Causal Inference in Statistics.* Wiley.

Pettey, C. (2017, Sept 18). *Prepare for the Impact of Digital Twins.* Retrieved from Gartner: https://www.gartner.com/smarterwithgartner/prepare-for-the-impact-of-digital-twins

Rafferty, J., Nugent, C., Chen, L., & Liu, J. (2015). Goal Lifecycles and Ontological Models for Intention Based Assistive Living within Smart Environments. *Computer Systems Science and Engineering, 30*(1).

Rafferty, J., Nugent, C., Liu, J., & Chen, L. (2017). From Activity Recognition to Intention Recognition for Assisted Living Within Smart Homes. *IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, 47*(3), 368-379.

Rafferty, J., Nugent, C., Liu, J., & Chen, L. (2017). From Activity Recognition to Intention Recognition for Assisted Living Within Smart Homes. *IEEE Transactions on Human-Machine Systems, 47*(3), 368-379.

Ravdin, L. (2008). Cognitive Disorders. *International Encyclopedia of Public Health*, 759-765.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection.* University of Washington, Allen Institute for AI, Facebook AI Research.

Sánchez, P., Fernández, P., Sánchez González, P., Charvill, J., Lofti, A., Langensiepen, C., . . . Aguilera, E. (2014). *iCarer Project: Intelligent Care Guidance and Learning Services Platform for Informal Carers of the Elderly.*

Savoy, A., Militello, L., Diiulio, J., Midboe, A. M., Weiner, M., Abbaszadegan, H., & Herout, J. (2019). Cognitive requirements for primary care providers during the referral process: Information needed from and interactions with an electronic health record system. *International Journal of Medical Informatics, 129*, 88-94.

Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., & Calinescu, A. (2020). *Digital Twins: State of the Art Theory and Practice, Challenges, and Open Research Questions.* Department of Computer Science, University of Oxford, Institute for Manufacturing, University of Cambridge.

Shi, L., Yifan, Z., Cheng, J., & Lu, H. (2019). Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. *CVPR*, 12026-12035.

Song, Y.-F., Zhang, Z., Shan, C., & Wang, L. (2020). Richly Activated Graph Convolutional Network for Robust Skeleton-based Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*.

Song, Y.-F., Zhang, Z., Shan, C., & Wang, L. (2020). Stronger, Faster and More Explainable: A Graph Convolutional Baseline for Skeleton-based Action Recognition. *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)* (pp. 1625–1633). New York, NY, USA: Association for Computing Machinery.

Suryadevara, N., Mukhopadhyay, S., Wang, R., & Rayudu, R. (2013). Forecasting the behavior of an elderly using wireless sensors data in a smart home. *Engineering Applications of Artificial Intelligence, 26*, 2641–2652.

Webb, J., & Ashley, J. (2012). Skeleton Tracking. In *Beginning Kinect Programming with the Microsoft Kinect SDK* (pp. 85-86). Apress.

WHO, Regional Office for Europe. (2017). *Age-friendly environments in Europe. A handbook of domains for policy action.* Denmark.

Wikipedia. (2021, May 28). Retrieved from Wikipedia, Visual programming language: https://en.wikipedia.org/wiki/Visual_programming_language

World Health Organization. (2015). *World report on ageing and health.* Geneva: WHO.

World Health Organization. (2016). *Diagnostic Errors: Technical Series on Safer Primary Care.* Geneva: WHO.

World Health Organization. (2017). *Global strategy and action plan on ageing and health.* Geneva: WHO.

World Health Organization. (2018, Feb 05). *Ageing and health.* Retrieved from World Health Organization: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health

(2020). *World Population Ageing 2019.* United Nations, Department of Economic and Social Affairs, Population Division.

(2020). *World Population Ageing 2020 Highlights: Living arrangements of older persons.* United Nations, Department of Economic and Social Affairs, Population Division.

(2019). *World Population Prospects 2019: Highlights.* United Nations, Department of Economic and Social Affairs, Population Division.

Xiao, Q., & Song, R. (2018). Action recognition based on hierarchical dynamic Bayesian network. *Multimedia Tools and Applications, 77*, 6955–6968.

Yan, S., Xiong, Y., & Lin, D. (2018). Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. *AAAI Conference on Artificial Intelligence.*

Yared, R., Abdulrazak, B., Tessier, T., & Mabilleau, P. (2015). Cooking risk analysis to enhance safety of elderly people in smart kitchen. *8th ACM International Conference on PErvasive Technologies Related to Assistive Environments, PETRA 2015 - Proceedings* (pp. 1-4). Association for Computing Machinery, Inc.

Yun, Y., & Yu-Hua Gu, I. (2017). Visual Information-Based Activity Recognition and Fall Detection for Assisted Living and eHealthCare. In C. Dobre, C. Mavromoustakis, N. Garcia, R. Goleva, & G. Mastorakis, *Ambient Assisted Living and Enhanced Living Environments* (pp. 395-425). Elsevier Inc.

ACKNOWLEDGEMENTS

Grazie alle persone che mi hanno accompagnato durante questo percorso. A chi mi ha sostenuto a costo di sacrifici. A chi ha sempre creduto nelle mie qualità. A chi ha riso e superato ostacoli al mio fianco. A chi condividerà con me il prossimo futuro.